Multi-Agent-Based Production Planning and Control

Jie Zhang

Shanghai Jiao Tong University
China

WILEY

国防工業出版社
National Defense Industry Press

Library of Congress Cataloging-in-Publication Data

Names: Zhang, Jie, 1963 September 21– author.
Title: Multi-agent-based production planning and control / Jie Zhang, Shanghai Jiao Tong University, China.
Description: First edition. | Hoboken, NJ, USA : John Wiley & Sons, Inc., [2017] | Includes bibliographical references and index.
Identifiers: LCCN 2016044467 (print) | LCCN 2017002169 (ebook) | ISBN 9781118890066 (cloth) | ISBN 9781118890080 (pdf) | ISBN 9781118890097 (epub)
Subjects: LCSH: Production planning. | Production control.
Classification: LCC TS155 .Z4349 2017 (print) | LCC TS155 (ebook) | DDC 658.5–dc23
LC record available at https://lccn.loc.gov/2016044467

Set in 10/12pt Warnock by SPi Global, Pondicherry, India

Contents

Preface

With the rapid development of advanced manufacturing technology, manufacturing models have developed: from single-piece production, mass production, small batch production with large product variation to customized production. More challenges lie ahead in this global manufacturing era, such as rapidly changing consumer demands, increased product varieties and shortened product life cycles and increasingly fluctuating markets, to name a few. Traditional push or pull production management methods have become more and more unsuited to the dynamic environment. In order to be more efficient in such an environment, flexibility, intelligence and self-adaptation have become the rule-of-thumb criteria for the evolution of new manufacturing systems. Therefore, a new hybrid push-pull production planning, scheduling and control system has been proposed.

Since 1992, Agent technology has gradually become a hot topic for Japanese and American research. In 1992, the Japanese Intelligent Manufacturing System Program concentrated on the invention of new Agent-based manufacturing methods as one of its main research areas. In 1993, the U.S. National Center for Manufacturing Science started a number of projects related to Agent-based manufacturing. Agent technology has been taken into consideration as a promising technique to solve production planning, scheduling and control problems in complex manufacturing systems so as to effectively enhance system flexibility, to improve product quality and to reduce production costs.

I have worked on investigating theories and techniques of production planning, scheduling and control in advanced manufacturing systems. In particular, I have completed National Natural Science Foundation Programs of China and National High Technology Research and Development Programs of China based on Agent technology. With the support of these projects, I have published a large number of papers in the field of Agent technology. This book is a systematic summary of these research results. The focus of this book is on Agent-based adaptive, intelligent, collaborative methods and technologies related to production planning, scheduling and control systems. The book also presents data acquisition systems based on RFID technology and OPC technology.

I am grateful to Xiaoxi Wang, Wei Qin, and Qiong Zhu for their assistances in the preparation of this book. Meanwhile, Cong Pan, Junliang Wang, Peng Zhang and Jungang Yang completed many auxiliary works. Lihui Wu, Gong Zhang, Shiyong Tian, Yijun Dong, Lei Sun, Guobao Liu and Zhi Xia have provided relevant documents, I thank all of them. I wish to acknowledge a large number of references in the completion of the manuscript. The responsibility is mine alone for any errors.

The writing of this book has been supported by the National Nature Science Foundation of China under Grant No. 51435009, Grant No. 51275307, and Grant No. 50875172, and by the National High Technology Research and Development Program (863 Program) of China under Grant No. 2007AA04Z019.

Theories, methods and applications of production planning, scheduling and control in modern manufacturing systems are rapidly developing. Agent technology has become a hot topic in the field of production planning, scheduling and control. If you have any questions about shortcomings and mistakes of this book, please do not hesitate to contact me.

Jie Zhang
December 2015

About this book

This book introduces methods and technologies of Agent-based production planning, scheduling and control on the basis of Job Shop manufacturing systems and Re-entrant manufacturing systems. It consists of eight aspects as follows:

1) Multi-Agent-based hybrid push-pull production planning and control framework
2) Multi-Agent-based production planning in distributed manufacturing systems
3) Multi-Agent-based production scheduling in Job Shop manufacturing systems
4) Multi-Agent-based production scheduling in Re-entrant manufacturing systems
5) Multi-Agent-based production control
6) Multi-Agent-based material data acquisition with RFID
7) Multi-Agent-based equipment data acquisition with OPC
8) Multi-Agent-based production planning and control prototype system

The purpose of this book is to track and trace the real-time production data, and to make real-time decisions in the production scheduling and control process.

The book is intended primarily for academic researchers in Agent-based manufacturing, and industry managers willing to develop a new manufacturing management model. This book is also a textbook and reference book for graduates and last-year undergraduates in mechanical engineering, industrial engineering, management, automation, and computer engineering and so on.

1

Agent Technology in Modern Manufacturing

1.1 Introduction

With the development of internet, computer, management, and manufacturing technologies, the manufacturing industry is undergoing a huge transformation from traditional manufacturing to agile manufacturing, networked manufacturing, virtual manufacturing, service-based manufacturing, and cloud manufacturing. These new manufacturing systems are characterized by smartness, integration, and flexibility, and can be well described as Agent technology. The cooperation and communication of multiple agents can be adopted to improve the performance of manufacturing systems.

1.2 Agent and Multi-Agent System

Research and application of Agent technology stem from a series of studies on distributed artificial intelligence conducted by MIT researchers in the 1970s.[1] Distributed artificial intelligence mainly focuses on solving distributed agent problems. There are two important branches:[2] distributed problems and Multi-Agent Systems (MASs). The distributed problems were conducted at an early stage in the distributed artificial intelligence area. The distributed problems have been extended to Multi-Agent Systems. The Multi-Agent System is a system with Agents of different abilities to complete collaboratively certain tasks or achieve certain objectives.[3–5]

1.2.1 Agent

The concepts, properties, and research methods of Agent technology are developed from artificial intelligence. It is difficult to define either artificial intelligence or Agent. Many different definitions have been given by different schools for different requirements. The earliest concept of Agent was defined based on the concurrent actor model proposed by Hewitt in the early 1970s.[6] In the concurrent actor model, Hewitt defined a term—*actor* with the characteristics of self-organization, interaction, and parallel execution. The most classic and widely accepted definition was given by Wooldridge, et al.[7] The definition contains "weak definition" and "strong definition". The weak definition defines an Agent as a hardware and software system with autonomous ability, social skill, and responsive and predictive ability; the strong definition includes the properties of the weak definition and also the properties of knowledge, mobility, veracity, rationality, and so on.

Computer science researchers[8] consider that an Agent is a computer system based on software and hardware; it also has autonomy reactivity, socialability, proactiveness, and other properties. From the perspective of the evolution of software design methods, agent-based software engineering methods are proposed on the basis of object-oriented software engineering methods. Moreover, decomposition and abstraction methods of complex software systems, distributed computing capabilities, interactive coordination mechanism, calculation model, and software architecture have been proposed.

Researchers in artificial intelligence are more inclined to a narrow point of view, except for the above properties. It is therefore necessary to give a more specific meaning for an Agent. Terms such as *belief*, *intention*, and *commitment* are used to describe an Agent. An Agent tries to mimic a human's thinking and intelligent behavior: for example, what the Agent is doing, what the Agent knows, what the Agent wants, and so on. This definition is developed on the basis of AI knowledge symbols. Shoham[9] thought an Agent was a symbolic reasoning system, which contained the expression of symbols on environment and expected behavior.

Therefore, an Agent is an intelligent individual. Wooldridge and Jennings[7] proposed that an Agent should have four basic attributes: autonomy, reactivity, social ability, and initiative. Sargent[10] considered that the most basic attributes of an Agent were reactivity, autonomy, goal-orientation, and environmental resistance. An Agent was defined by Muller[11] as follows: 1) it is necessary to have other Agents and a virtual world where an Agent exists; 2) an Agent can perceive a virtual world and influence the virtual world; 3) an Agent can at least partly represent the virtual world; 4) an Agent is target-oriented and has the ability to arrange its own activities; 5) an Agent can communicate with other Agents. Most researchers think that an Agent should not only meet basic properties, but should also have other properties according to application requirements: for example, mobility, learning and adaptability, interactivity, planning ability, rationality, persistent or time continuity, and so on. Three directions of current research are intelligence, agency, and mobility.[12] From the intelligence point of view, an Agent is an expert system; *agency* means that an Agent can be used to represent the role of a man and machine; while mobility means that an Agent can move or run on a different machine on the internet.

As the previous presentation demonstrates, an Agent should have the following properties:[13–21]

1) *Autonomy:* An Agent can control its behavior and internal state by itself, and it cannot be controlled by others. This is used to differentiate an Agent with other concepts such as process and object.
2) *Reactivity:* An Agent can feel the environment and respond appropriately to environment-related events.
3) *Sociality:* An Agent is in a social environment constituted by multiple Agents. These Agents exchange information with each other in some interactive methods. These Agents collaborate with each other to solve different problems and help other Agents complete related activities. Agents exchange information by a communication language.
4) *Initiative:* The reaction of an Agent to the environment is a goal-directed initiative behavior. In some cases, the behavior of the Agent is triggered by its own requirements. The reactive behavior is a kind of positive behavior or an active communication with the environment.

5) *Adaptability:* An Agent can respond to environmental changes, adopt a goal-oriented action at the appropriate time, and learn from its own experience, the environment, and the interaction process with other Agents.
6) *Interoperability:* An Agent can work with other Agents to complete complex tasks, which is a social behavior.
7) *Learning ability:* An Agent can learn from the surrounding environment and cooperative experiences so as to improve its own capability.
8) *Evolutionary development:* An Agent can improve itself through learning, and reproduce and follow Darwin's natural selection rule "survival of the fittest".
9) *Honesty:* An Agent does not intend to deceive users.
10) *Rationality:* the action taken by an Agent and its consequences will not harm its own interest and other Agents' interests.
11) *Persistence:* An Agent is ongoing, not temporary, its status should be consistent, which is not in contradiction with property (8).
12) *Mobility:* An Agent should have the ability to move independently in the network, while its status remains unchanged.
13) *Reasoning:* An Agent can reason and forecast in a rational manner according to accumulated past knowledge, states of the current environment, and other Agents.
14) *Others*: philanthropic, adventurous or conservative, helpful or hostile, and so on.

The above attributes show that an Agent is similar to a person, which provides a new method for solving complex problems in computer science and artificial intelligence. Although an Agent may have a variety of properties, researchers and developers do not need to develop one Agent or an Agent system with all the attributes. Agents with several attributes and Multi-Agent systems with several attributes are developed according to actual requirements.

1.2.2 Multi-Agent System

Agent Systems can be classified into two classes: Single-Agent Systems and Multi-Agent Systems (MASs). The research of a Single-Agent System focuses on simulating human intelligent behavior; it concentrates on investigating human intelligent behavior such as computing ability, reasoning ability, memory, learning ability and intuition, and so on. The research of a MAS focuses on the collaborative process among autonomous intelligent Agents that generate their corresponding behaviors or solve problems by coordinating Agent goals and planning Agents. In the problem-solving process, these Agents share all their knowledge about related problems and methods in order to achieve a global objective, or their own local objectives.[22–25]

As regards a MAS, a computing system aims to complete collaboratively certain tasks or achieve objectives by some Agents. The system consists of multiple autonomous or semi-autonomous Agents.[26] In a MAS, each Agent cooperates with other Agents to complete a complex task that cannot be solved by single Agent. All the Agents are autonomous, running in a distributed mode, or even heterogeneous. The subroutine, function, or process of each Agent are different, its goal and behavior are relatively autonomous and independent. Each Agent cooperates with other Agents in order to deal with conflict among them. A MAS has advantages of traditional distribution concurrent problem, and it runs in an interactive communication mode. Compared with a single Agent, each Agent in a MAS has incomplete information and it is able to solve problems, the data is dispersed or distributed, and the computing process is asynchronous, concurrent, or parallel. A MAS is very suitable to express an environment with a variety of methods and entities. A MAS has the following features:

1) *Sociality*. In a MAS, an Agent may be in a social environment constituted by several Agents and may have the information and knowledge from other Agents. Agents communicate with each other by using a special language to complete the cooperation and negotiation activities. For example, in an Agent-based manufacturing process production management system, Multi-Agents representing the roles of customers, sales, production management, material procurement and quality inspection departments cooperate together to complete production tasks.
2) *Autonomy*. In a MAS, when an Agent sends out a service request, other Agents that have this service ability and interest will accept it. One Agent cannot force another Agent to provide service.

3) *Cooperation*. In a MAS system, Agents with different objectives work collaboratively to solve the problem through mutual cooperation and negotiation. The coordination process consists of resource sharing coordination, the producer/consumer relationship collaboration, tasks/subtasks relationship collaboration.

The Multi-Agent System theory is developed on the basis of the Single-Agent model and structure, which focuses on investigating interoperability, consultation and cooperation among Agents on the basis of the Single-Agent theory. The consultation and collaboration activities in a MAS are realized based on social organization theory and modeling and implementing theory. The social organization theory provides a society-oriented conceptual model about integration, interaction, communication and collaboration; while the modeling and implementing theory is used to eliminate the gap between the society-oriented conceptual model and the reality. Therefore, the process to develop a MAS consists of the following aspects:[27–30]

1) *Agent model*. An Agent model is developed in order to meet the requirements of individual autonomy, group interaction and the environment. The organizational structure, knowledge composition and operation mechanism of an Agent are described in a certain level of abstraction.
2) *MAS architecture*. The asynchronous, consistency, autonomy and self-adaptive features of an Agent are affected by selecting an architecture. This will determine information transmission channels and transmission ways for a single Agent internal intelligence collaborative behavior.
3) *Interaction and communication*. The interaction activity is a basic requirement for multiple Agents to collaborate with each other. The communication activity is the basis for the interaction activity. The communication activity includes two aspects: the first is to construct the underlying communication mechanism, and the second is to construct or select an Agent communication language.
4) *Consistency and collaboration*. Consistency describes the overall features of distributed artificial intelligence systems. Collaboration expresses the behavior and interaction patterns among agents. Good collaboration is important to

achieve the stability and consistency of the system's overall behavior. An efficient MAS should trend toward the overall consistency quickly through less learning.

5) *MAS planning*. MAS planning activity is a kind of adaptation planning activity, which reflects the continuous changing process of the environment.[31]
6) *Conflict management*. Conflict in the collaborative process is very common. Conflict can be classified into three classes: resource conflict, objective conflict, and result conflict.[32]

1.3 Agent Technologies in Manufacturing Systems

The development of Agent technology borrows many ideas and technological achievements from computer science, sociology, organizational science, economics and ecology and other disciplines, which has obvious advantages in many ways: extensive adaptability for real-world applications, simplicity of design and good systematicness and others.[33, 34] Agent technology has been widely applied in manufacturing, communications, air traffic control, traffic management, information management systems, business process management, remote diagnostics and education and entertainment and many other fields,[35–39] which had achieved remarkable results. This section focuses on introducing Agent technology application in modern manufacturing.

1.3.1 Contemporary Manufacturing Systems

The third industrial revolution through the development and usage of computer technologies is affecting the manufacturing industry in a serious way. From the early 1970s, several influencing and representative manufacturing modes and relevant technologies had been presented, for example, the Toyota JIT system in 1980s, the Agile manufacturing system and the networked manufacturing system that were famous in 1990s. These advanced manufacturing modes reflect changes in different periods of external demand, manufacturing models, and related technical support, and reflect the evolution process: "information integration – process integration – inter-enterprise integration".

In this section, we will introduce the current situation and trends of the manufacturing industry, present a literature review concerning production planning and control technologies, and analyze problems and solutions.

Supported by the US Congress and the Defense Department of United States in 1991, the Iacocca institute and the other 13 companies prepared a "21st Century Manufacturing Enterprise Strategy" report, which proposed a new strategy – agile manufacturing to revitalize the manufacturing sector. As a new manufacturing model, agile manufacturing quickly got the recognition and support of US industry, government agencies and the community, and soon became a theoretical research and manufacturing practice hotspot.

Virtual enterprise is an application mode of agile manufacturing. A virtual enterprise is a temporary alliance that comes together to share skills or core competencies and resources in order to respond rapidly to business responsibilities. The whole cooperation is supported by computer networks. Most of the research for virtual manufacturing focuses on organizational management, cooperative partner selection, profit assignment problems, and so on.

From 1996–2000, funds became available to support the agile manufacturing program, for example, the national industrial information system program of United States, the virtual manufacturing network program of Russia and the United States, the European information technology development program, the Japanese smart manufacturing system plan. These research projects focus on investigating virtual enterprises that run related enabling technologies, commercial operation infrastructure, collaborative design, and information integration platforms. The Ecolead program formulated a cooperative organization network including 28 units in 15 countries.

1.3.2 Agents in Production Planning and Control Systems

A production planning and control system is the core and key technology of production management systems. An excellent production planning and control system is an important tool to improve the overall automation level of enterprises and provide significant economic benefits for enterprises. A production-planning and control system can directly determine whether the manufacturer can complete specific tasks in accordance with the expected demand. Its core function is to manage production tasks and resource allocation and utilization in manufacturing systems, and to meet customers' demands in the best possible way. A production planning and control system should contain the following processes: decomposing product tasks, analyzing resource demands, determining the operation sequence of a job, allocating machines, and monitoring real-time task progress. Meanwhile, this system should be able to deal with sudden changes in the actual manufacturing environment, such as lack of material, random machine breakdown, order changeover and rush orders, and so on. Even though production planning and control problems are complex, Agent technology has been introduced in this field. A MAS has a certain adaptability in a dynamic environment; it can independently adjust the behavior of individuals in order to respond rapidly to sudden changes in manufacturing systems.

Parunak, et al.[40] developed a production planning and shop floor control system based on Agent technology. Shen et al.[41] studied the integrated modeling framework for business-oriented mixed Agents. Lin and Solberg[42] proposed an autonomous Agent-based integrated production planning and control framework. In their study, a general methodology that consists of a market-based model, a job priority strategy, a multi-stage negotiation technology were developed to adapt to changes in the manufacturing environment. Hadavi et al.[43] proposed a Multi-Agent distributed dynamic planning, scheduling and control system. Applications of MASs to production planning and control problems were summarized by Maria and Sergio.[44] It was noted that its related research tended to be more diverse. Baker[45] studied a MAS-based shop scheduling algorithm. Shaw[46] proposed Agent-based production scheduling and control strategies. In his study, a manufacturing unit could subcontract its task as a subcontract to other manufacturing units by using a bidding mechanism. In the study of Wang et al.,[47] the Agent technology was used to solve real-time distributed intelligent manufacturing control problems. Wiendahl et al.[48] studied a self-organizing production control system based on Agents.

Butler[49] proposed a Multi-Agent system architecture to solve distributed dynamic scheduling problems. In his study, the scheduling process was divided into two levels: the first layer was used to assign manufacturing units to jobs by using an Agent-based consultation mechanism, the second layer was adopted to allocate dynamically shared manufacturing resources. Shen and Norri[50] proposed a hybrid Agent system architecture to solve scheduling and rescheduling problems.

Many Chinese researchers have also proposed different MAS solutions for production planning and control systems. In the study by Zhang Jie, Li Penggen et al.,[51] virtual manufacturing cells were introduced to deal with production planning and control process. The system consisted of a shop floor layer, a virtual cell layer and a resource layer. On the basis of this study, some MAS-based production planning and control solutions have been developed for solving Job Shop,[52–60] reentrant Manufacturing System,[61–66] agile manufacturing systems,[67–72] and other planning and control problems for complex manufacturing systems. Zhu Qiong, et al.[73] proposed a Multi-Agent-based collaborative negotiation mechanism for solving dynamic Job Shop scheduling problems. Zeng Bo, Yang Jianjun, et al.[74, 75] proposed a Job Shop scheduling and control system that hybridizes a MAS-based Generalized Partial Global Planning (GPGP) mechanism with a Task Analysis, Environment Modeling and Simulation (TAEMS) language. Liao Qiang et al.[76] proposed a Multi-Agent-based Job Shop dynamic scheduling model by using field bus. Gao Guojun et al.[77] used Agent technology to developed a reconfigurable enterprise information system by using Agent technology. Liu Jinkun et al.[78] proposed an Agent-based steel industrial production process control system.

1.3.3 The Existing Requirements

An Agent system can be used to coordinate production planning dynamically and control activities, adjust individual behavior, and respond rapidly to deal with product changes, machine breakdown, and other incidents. Therefore, Agent technology is suitable for the production planning and control process in manufacturing systems to make decisions. Agent-based production planning and control methods and techniques are presented in detail in this book. A Multi-Agent-based hierarchical adaptive, intelligent, collaborative production planning and control system is developed to investigate some Multi-Agent-based planning, scheduling, and control problems. Moreover, a MAS-based data acquisition technology is proposed for complex production processes in order to collect real-time data and track real-time production processes. The MAS-based technology provides a set of optimal solutions and ideas for the production planning and control process in manufacturing systems.

1.4 Book Organization

1.4.1 Purpose of the Book

Production planning, scheduling, and control optimization have become urgent demands and a trend in modern complex manufacturing systems such as weapon manufacturing systems and semiconductor manufacturing systems. In this book, weapon manufacturing systems are characterized as small-batch manufacturing systems, which are similar to typical Job Shop Manufacturing systems. The semiconductor manufacturing systems are characterized as typical re-entrant manufacturing systems. The book will focus on developing a hybrid push-pull production planning and control system architecture based on MASs to describe the characteristics of Job Shop and re-entrant manufacturing systems. The overall system objectives are completed by communicating and collaborating amongst Agents to manage and make decisions so as to respond rapidly to internal and external changes in the manufacturing environment. The production planning and control process presented in this book consists of three layers: the production planning layer, the production scheduling layer, and the production control layer. In the production planning layer, MAS-based production planning methods for distributed manufacturing systems are given. In the production scheduling layer, a Multi-Agent double feedback strategy-based scheduling method is developed for Job Shop Manufacturing systems, and a Multi-Agent hierarchical adaptive production scheduling method is proposed for Re-entrant manufacturing systems; in the production control layer, the radio frequency identification (RFID) technology and OLE for Process Control (OPC) technology and Multi-Agent Systems are integrated, and a material and equipment data acquisition method is then designed for manufacturing systems. This method is able to collect heterogeneous device data and integrate information between heterogeneous networks. This method also provides a new way for tracking real-time production processes in complex manufacturing systems and a foundation for real-time production decision-making in manufacturing systems.

1.4.2 Scope of the Book

This book systematically presents methods and technologies concerning Agent-based production planning systems. The content of this book is illustrated in Figure 1-1. Chapter 1 and Chapter 2 introduce advantages and applications of Agent technology. Chapter 3 presents requirements of production planning and control systems. An Agent-based push-pull production planning and control system is developed. Agent-based production planning and control technologies for distributed production systems are introduced in Chapter 4 to Chapter 9. Chapter 4 proposes a Multi-Agent contract net protocol and bid auction protocol based production planning approach for distributed production systems. Chapter 5 develops a Multi-Agent double feedback–strategy based production scheduling method for Job Shop production systems. Chapter 6 proposes a Multi-Agent hierarchical adaptive production scheduling architecture for Re-entrant manufacturing systems. Chapter 7 presents a Multi-Agent production control system. Chapters 8 and 9 present data acquisition technology based on RFID technology and OPC technology, and construct a Multi-Agent material data acquisition system and a Multi-Agent equipment data acquisition system. Chapter 10 presents the prototype of an Agent-based production planning and control system.

1.4.3 Content of the Book

Chapter 1 introduces Agent technologies in modern manufacturing. A short review concerning Agent technology is given to provide the background for investigating production planning, scheduling, and control approaches.

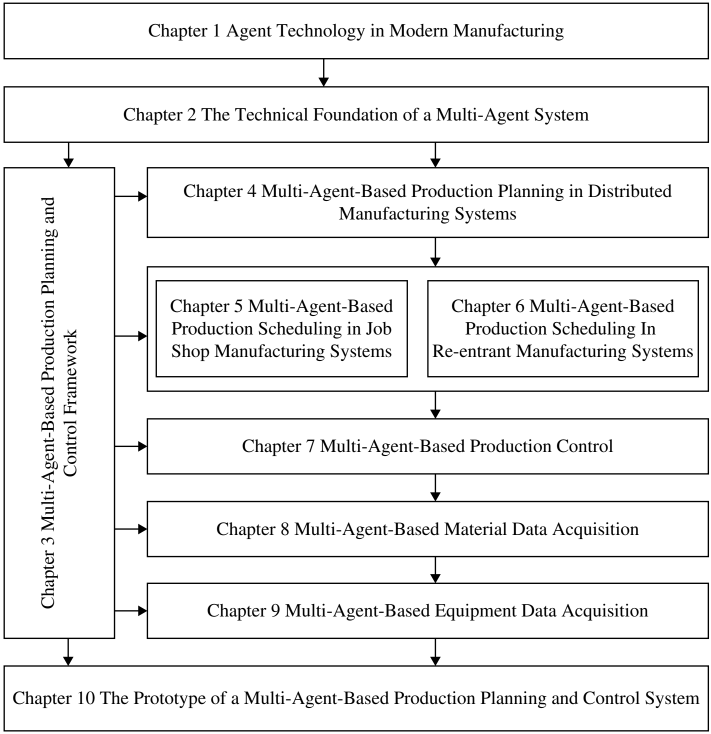

Figure 1-1 The content of this book.

Chapter 2 presents Agent technologies used in this book, which includes three aspects, that is the structure of an Agent and a Multi-Agent System, the interaction model of a Multi-Agent System, and communication protocols and interaction protocols of Multi-Agent Systems.

Chapter 3 analyzes production planning and control activities and operation modes, and requirements of a production planning and control system. A hybrid Agent-based push-pull production planning and control system is developed.

Chapter 4 presents Agent-based production planning methods for distributed manufacturing systems. First, the production planning process in distributed manufacturing systems is investigated. Second, a production planning model and a MAS structure are developed. Third, a contract net protocol–based

MAS collaborative production planning method is proposed in which plants collaborate in a distributed manufacturing system with complete information sharing. Finally, a bidding auction protocol–based MAS collaborative production planning method is proposed where plants collaborate in a distributed manufacturing system with incomplete information sharing.

Chapter 5 presents Agent-based production scheduling methods for Job Shop manufacturing systems according to the basic principles of push and pull modes. In particular, a multi-Agent dual feedback–based production scheduling method is developed. Then, a hierarchical optimization theory–based positive feedback job scheduling method, and an ant colony negotiation mechanism–based negative feedback rescheduling method are proposed.

Chapter 6 develops a Multi-Agent-based hierarchical adaptive production scheduling architecture to describe the characteristics of re-entrant manufacturing system. A combinatorial auction–based method is developed in the system layer; a GPGP-CN based method is developed for hierarchical production scheduling processes in the machine layer; and a fuzzy neural network–based adaptive rescheduling method is developed for re-entrant manufacturing systems.

Chapter 7 proposes a Multi-Agent-based production control method by analyzing requirements of production control activities. Several important business Agents and related methods in the production control process are presented.

Chapter 8 presents the basic concepts and function requirements of material data acquisition, and develops a Multi-Agent RFID technology–based material data acquisition system.

Chapter 9 presents a Multi-Agent OPC technology based equipment data acquisition system.

Chapter 10 presents both hardware architecture and software architecture of an Agent-based production planning and control prototype system.

References

[1] Leon, S., and Kuldar, T. *The Art of Agent-Oriented Modeling.* Cambridge, MA: MIT Press, 2009.

[2] Shi, Z. *Advanced Artificial Intelligence*. Beijing: Science Press, 1998, 1.

[3] Minsky, M. *The Society of Mind*. New York: Simon & Schuster, 1986, 1.
[4] Minsky, M., and Riecken, D. A Conversation with Marvin Minsky about Agents. *Communications of the ACM*, 1994, 37(7): 23–29.
[5] Mare, H. Debugging Multi-Agent System. *Information and Software Technology*, 1995, 37(2): 102–112.
[6] Hewitt, C. A Universal Modular ACTOR Formalism for AI. In *Proceedings of the 3rd International Joint Conference on Artificial Intelligence*, Stanford, CA, 1973.
[7] Wooldridge, M., and Jennings, N. Intelligent Agents: Theory and Practice. *The Knowledge Engineering Review*, 1995, 10(2): 115–152.
[8] Finin, T., Fritzson, R., Mekay, D., MeEntire, R. KQML-A Language and Protocol for Knowledge Department and Information Exehange. Technical Report CS-94-02, Computer Science. University of Maryland, 1994.
[9] Shoham, Y. Agent Oriented Programming. *Artificial Intelligence*, 1993, 60: 51–92.
[10] Sargent P. Back to School for a Brand New ABC. *The Guardian*, 1992, 12(3): 12–28.
[11] Muller, J. P., Wooldridge, M., and Jennings, N. R. *Intelligent Agents,* III. LNAI, Berlin, Springer, 1996.
[12] Wang, R., and Zhou, C. Mobile Agent Revie. *Computer Application Research*, 2001, 6: 9–11.
[13] Franklin, S., and Graesser, A. Is It an Agent, or Just a Program? A Taxonomy for Autonomous Agents. *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*, London: Springer-Verlag, 1996.
[14] Keil, F. C. *Concepts, Kinds, and Cognitive Development*. Cambridge, MA: MIT Press, 1989.
[15] Charles, J. P. Agent-Based Engineering, the Web, and Intelligence. *IEEE Expert/Intelligent Systems & Their Applications*, 1996, 11(6): 24–29.
[16] Gilbert, D. *IBM Intelligent Agent Strategy*. New York: IBM Corporation, 1995.
[17] Maes, P. Artificial Life Meets Entertainment: Life Like Autonomous Agents. *Communications of the ACM*, 1995, 38(11): 108–114.
[18] Wang, H. Agent Technology and Distributed Computing. *Computer World*, 1998, 4.

[19] Bai, S. Intelligent Agent Strategy. *Computer World*, 1998, 4.
[20] Nwana, H. S. Software Agents: an Overview. *The Knowledge Engineering Review*, 1996, 11(3): 205–244.
[21] Yang, K., Zhai, Y. and Liu, D. Agent: Classification and Characteristics. *Computer Science*, 1999, 26(9): 30–34.
[22] Weiming, S. and Douglas, H. N. A Hybrid Agent-oriented Infrastructure for Modeling Manufacturing Enterprises. http://ksi.cpsc.ucalgary.ca/KAW/KAW98/shen/.
[23] Tan, G. W., Hayes, C. C., and Shaw, M. An Intelligent-agent Framework for Concurrent Product Design and Planning. *IEEE Transaction on Engineering Management*, 1996, 43(3): 297–306 .
[24] Sikora, R., and Shaw, M. J. Coordination Mechanisms for Multi-agent Manufacturing System: Applications to Integrated Manufacturing Scheduling. *IEEE Transaction on Engineering Management*, 1997, 44(2): 175–187.
[25] Martin, W., Moira, C. N., and Werner, S. Multidatabase Agents for CIM Systems. *International Journal of Computer Integrated Manufacturing*, 1996, 9(4): 293–298.
[26] Jennings, N. R., Sycara, K., and Wooldridge, M. A Roadmap of Agent Research and Development. *Autonomous Agents and Multi-Agent Systems*, 1998, 10(2): 199–215.
[27] Zhong, S., Liu, D., and Chen J. Open, Self-adaptive, Distributed Agent System. *Natural Science of Jilin University*, 1996, 1: 35–42.
[28] Mare, H. Debugging Multi-agent System. *Information and Software Technology*, 1995, 37(2): 102–112.
[29] Bouchefra, K., Auge, P., Maurin, T., et al. Multi-agent Based Architecture Specification and Verification. http://ksi.cpsc.ucalgary.ca/KAW/KAW98/bouchefra/
[30] Stephen, E. C. New Technologies Mean New Opportunities and Challenges. *Intelligent System & Their Application*, 1996, 11(6): 1–2.
[31] Liu, D., Yang, K., and Che, J. Review and Trend Analysis for Agent Technology. *Chinese Journal of Software*, 2000, 11(3): 315–321.
[32] Liang, Q., Xu, X., and Zhang, Z. Cooperation and Control Mechanism in MAS. *Control and Decision*, 1996, 11(5): 522–537.
[33] Wang, D. "Adaptation of Agile Manufacturing Unit Management and Control System." PhD Thesis, Shanghai Jiao Tong University, 2001, 5.

[34] Jeff, Y., and Jay, M. An Intelligent Agent Framework for Enterprise Integration. *IEEE Transactions on Systems, Man, and Cybernetics*, 1991, 21(11): 1391–1407.

[35] Song, D., and Mai, Z. Agent based DBPM/AB Design and Development. *Computer Science*, 2000, 27(3): 35–39.

[36] Minsky, M., and Riecken, D. A Conversation with Marvin Minsky about Agents. *Communications of the ACM*, 1994, 37(7): 23–29.

[37] Mare, H. Debugging Multi-agent System. *Information and Software Technology*, 1995, 37(2): 102–112.

[38] Hu, S., Zhang, L., and Zhong, Y. MAS Theory, Technology and Application. *Computer Science*, 1999, 26(9): 20–24.

[39] Shi, Z. *Advanced Artificial Intelligence*. Beijing: Science Press, 1998, 1.

[40] Parunak, V. D., Baker, A., and Clark, S. The AARIA Agent Architecture: From Manufacturing Requirements to Agent-Based System Design. In *Working Notes of the Agent-Based Manufacturing Workshop*. Minneapolis, MN, 2001, 5.

[41] Shen, W., and Norrie, D. H. A Hybrid Agent-oriented Infrastructure for Modeling Manufacturing Enterprises. *Proceedings of KAW'98*. Banff, Canada, 1998.

[42] Lin, G., and Solberg, J. Integrated Shop Floor Control Using Autonomous Agents. *IIE Transaction*, 1992, 24(3): 57–71.

[43] Hadavi, K., Hsu, W. L., and Volgt, K. ReDS-a Dynamic Planning, Scheduling, and Control System for Manufacturing. *Journal of Manufacturing Systems*, 1990, 32(4): 332–344.

[44] Maria, C., and Sergio, C. Multi-Agent System in Production Planning and Control: An Overview. *Production Planning and Control*, 2004, 15(2): 106–118.

[45] Baker, A. D. A Survey of Factory Control Algorithms That Can be Implemented in a Multi-agent Hierarchy: Dispatching, Scheduling and Pull. *Journal of Manufacturing Systems*, 1998, 17(4): 297–320.

[46] Shaw, M. J. Dynamic Scheduling in Cellular Manufacturing Systems: A Framework for Networked Decision Making. *Journal of Manufacturing Systems*, 1988, 7(2): 83–94.

[47] Wang, L., Balasubramanian, S., and Norrie, D. Agent-Based Intelligent Control System Design for Real-time Distributed Manufacturing Environments. In *Working Notes of the Agent-Based Manufacturing Workshop*. Minneapolis, MN, 1998.

[48] Wiendahl, H. P., and Ahrens, V. Agent-based Control of Self-organized Production Systems. *Annals of the CIRP*, 1997, 46(l): 365–368.

[49] Butler, J., and Ohtsubo, H. ADDYMS: Architecture for Distributed Dynamic Manufacturing Scheduling. *Artificial Intelligence Application in Manufacturing*. Cambridge, MA: MIT Press, 1992.

[50] Shen, W., and Norrie, D. H. An Agent-Based Approach for Dynamic Manufacturing Scheduling. *Workshop Notes of the Agent-Based Manufacturing Workshop at Autonomous Agents*, 1998.

[51] Zhang, J., Gao, L. and Li, P. *Agent Technology and Its Application in Modern Manufacturing*. Beijing: Science Press, 2004.

[52] Liu, S. P., Zhang, J., and Li, P. G. Study on Multi-Agent Based Shop Floor Manufacturing and Control System. *Proceedings of International Conference on Advanced Manufacturing Systems and Manufacturing Automation*, 19–21 June 2000, Guangzhou, China, 883–886.

[53] Liu, S. P., Zhang, J., et al. Agent Based Job Shop Planning and Scheduling. *Mechanical and Electronics*, 2001, (5): 28–31.

[54] Liu, S. P., Rao, Y. Q., Zhang, J., et al. Model of Production Planning and Scheduling in Shop Floor based on Collaboration and Competition. *Wuhan University Journal of Natural Sciences* (English Edition), 2002, 7(1): 77–81.

[55] Zhang, J., Gao, L., and Li, P. G. A Holonic Architecture of the Concurrent Integrated Process Planning System. *Journal of Materials Processing Technology*, 2003, 139(1–2): 267–272.

[56] Zhang, J., Shi, K., and Li, P. MAS-Based Job Shop Control. *Journal of Huazhong University of Technology*, 1998, 26(12): 4–7.

[57] Liu, S. P., Rao, Y. Q., Zhang, J., et al. Modeling and Implementation of a Distributed Shop Floor Management and Control System *Chinese Journal of Mechanical Engineering* (English edition). 2002, 15(3): 213–217.

[58] Chan, F. T. S., Zhang, J., He, J., et al. Investigation on Multi-Agent Based Shop Floor Control System. *International Congress on Intelligent Systems and Applications* (ISA'2000). The University of Wollongong, Australia, 12–15 December, 2000.

[59] Chan, F. T. S., and Zhang, J. A Multi-Agent Based Agile Shop Floor Control System. *International Journal of Advanced Manufacturing Technology*, 2002, 19: 764–774.

[60] Zhang, J., Liu, S., and Li, P. MAS Based Job Shop Re-engineering Model. *Chinese Journal of Mechanical Engineering*, 2000, 11(4): 432–434.
[61] Zhang, J., Wu, L., and Zhai, W. *Control of Re-Entrance Production System*. Beijing: Science Press. 2009, 1.
[62] Wu, L., and Zhang, J. MAS Based KCTPN Modeling Method for Wafer Fabrication Manufacturing. *Computer Integrated Manufacturing System*, 2009,15(10): 1921–1929.
[63] Zhang, J., Zhai, W. B., and Yan, J. Q. Multiagent-Based Modeling for Re-entrant Manufacturing System. *International Journal of Production Research*, 2007, 45(13): 3017–3036.
[64] Zhai, W. B. Zhang, J., Yan, J. Q., and Ma, D. Z. Agent-Oriented Colored Petri-Net Based Interactive Protocol Modeling Technologies of Semiconductor Fabrication Line. *Journal of Shanghai Jiaotong University*, 2005, 39(7): 11150–11554.
[65] Zhai, W. B., Chu, X. N., Zhang, J., et al. Research on AOCTPN-Based Modeling and Simulation Technology of Semiconductor Fabrication Line. *International of Journal Advanced Manufacturing Technology*, 2006, 28(7–8): 814–821.
[66] Zhai, W. B. Zhang, J., Yan, J. Q., and Ma, D. Z. Research on TEAEMS/GPGP-CN Based Dynamic Scheduling Technology of Semiconductor Fabrication Line. *Chinese Journal of Mechanical Engineering*, 2005, 41(3): 53–58.
[67] Li, P. G., and Zhang, J. *Reconfiguration and Control for Agile and Intelligent Manufacturing System*. Beijing: China Machine Press, 2002.
[68] Chan, F. T. S., and Zhang, J. Modeling of Agile Manufacturing Systems. *International Journal of Production Research*, 2001, 39(11): 2313–2332.
[69] Chan, F. T. S., and Zhang, J. Agent- and CORBA-based Application Integration Platform for an Agile Manufacturing Environment. *International Journal of Advanced Manufacturing Technology*, 2003, 21: 460–468.
[70] Li, S., Rao, Y., et al. Re-engineering of Agile Manufacturing. *High Tech Letter*, 2002, 12(5): 67–70.
[71] Shi, K., Zhang, J., et al. *Journal of Huazhong University of Technology.* 1999, 27(8): 21–23.

[72] Zhang, J., Chan, F. T. S., Li, P. G., et al. Investigation on the Reconfigurable Cell Control System for Agile Manufacturing. *International Journal of Production Research*, 2002, 40(15): 3709–3724.

[73] Zhu, Q., Chen, X., and Zhang, J. Multi-Agent Based Dynamic Scheduling Method. *Journal of Shanghai Jiao Tong University*, 2008, 42(7): 1046–1050.

[74] Zeng, B., and Yang, J. MAS based GPGN Jobshop Scheduling. *Journal of Mechanical Engineering*, 2001, 37(8): 80–83.

[75] Wei, W., Yang, J., et al. Multi-Agent Based Distributed Shop Floor Control System. *Mechanical Design and Mechanical Engineering*, 2001, 30(l): 28–33.

[76] Liao, Q., Zhou, K., and Zhang, B. Multi-Agent and Fieldbus Based Dynamic Job Shop Scheduling Problem. *Chinese Journal of Mechanical Engineering*, 2000, 11(7): 757–759.

[77] Gao, G., Duan, Y. and Zhang, S. CORBA and Multi-agent based Reconfigurable EIS. *Computer Integrated Manufacturing System*, 2000, 6(3): 25–29.

[78] Liu, J., and Wang, S. MAS for Steel Industrial Processing Management. *Information and Control*, 1998, 27(6): 427–432.

2

The Technical Foundation of a Multi-Agent System

2.1 Introduction

Since the 1990s, with the development of computer networks, computer and communications technology, Agent technology has not only become a hot spot for distributed artificial intelligence research, but also has become a hot field of information technology. It has been noted that reseach achievements in various fields of artificial intelligence should be integrated into an Agent of intelligent behavior, and more importantly, that the nature of artificial intelligence is social intelligence. "Collaboration", "competition" and "negotiation" are the main manifestations of human intelligent behavior. The prerequisite for the application of the Agent is to achieve its intelligent behavior by building an Agent, guiding its interaction with the surrounding environment, and communicating with other Agents, which plays an important role in forming new computing and problem-solving norms, and establishing an Agent technology–based distributed collaborative model with certain autonomy.

2.2 The Structure of an Agent

The basic function of an Agent is to interact with the external environment, to obtain and process the information obtained, and to influence the environment. An Agent can be seen as a black box, which senses the environment by sensors and acts on the environment through actuators. Most Agents not only interact

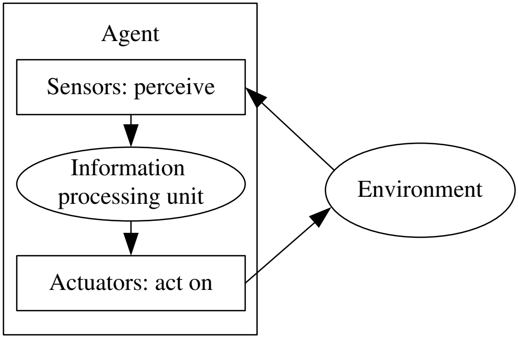

Figure 2-1 The basic structure of an Agent.

with the environment, but also process and interpret the information received for their own purposes.

The basic structure of an Agent is shown in Figure 2-1. An Agent can be defined as a series of operations that consist of information-perceiving action, information-processing action and the action on the environment. Assuming that O is the information collection sensed by the Agent at any time and A is the set of possible actions that can be completed by an Agent in the external world, then the information processing function f of an Agent is:

$$^{*}O \rightarrow A$$

The structure of an Agent is a specific method to construct an Agent, which illustrates how an Agent is decomposed into several modules and how these modules interact with each other. Modules and their interaction define how the Agent determines the output operations $\alpha(\alpha \subset A)$ in accordance with the perceiving information $o(o \subset O)$ obtained and its processing methods.

The main studies of the structure of an Agent include the composition of an Agent, their relations, and the mechanisms of perceiving and acting on the environment. For example, in terms of design, some structures of an Agent can be abstracted as a blackboard, in which sensors detect the environment information and record the information sequence on the blackboard, the information-processing unit receives information sequence to form an action sequence listed on the blackboard, the actuators act on the external environment, and the feedback information after the execution is received by the blackboard. In terms of the theoretical basis, the structure of an Agent can be classified into three major classes:

1) *Thinking Agent* (Cognitive or Deliberative Agent). In this type, the Information processing unit consists of a variety of behavioral knowledge, domain knowledge, and decision rule. The information-processing unit is able to implement complex logical reasoning in order to make decisions.
2) *Reactive Agent*. In this type, the information-processing unit doesn't include any domain knowledge or behavior knowledge of the environment. Complex reasoning mechanisms are not adopted by the information-processing unit, and decision-making is performed by using the predefined rules inside the information-processing unit.
3) *Hybrid Agent*. This is a new structure that is formed by the combination of two structures above.[1–4]

2.2.1 Thinking Agent

A thinking Agent contains the information model that can explicitly represent the environment characteristics, and its decision-making behavior and actions (such as what action to perform) are determined by means of logical reasoning and calculation. The structure is shown in Figure 2-2.

The cognitive components of a thinking Agent consist of sensors and actuators. It receives the information of the external environment by sensors, performs information fusion based on internal state, and generates the information parameter model, including external environment information and internal state information. Then the next plan is developed with the support of a knowledge base, a series of actions is generated by optimizing objective, and actions affect the environment by actuators.

The core of a thinking Agent is as follows: how to map the real environment into an accurate, appropriate expression formed by symbols and formulas within a certain time so as to develop a mathematical model; how to perform reasoning and decision-making by planning the information obtained within a certain time. Although many scholars have studied this area, it is hard to solve some simple problems such as common-sense reasoning.

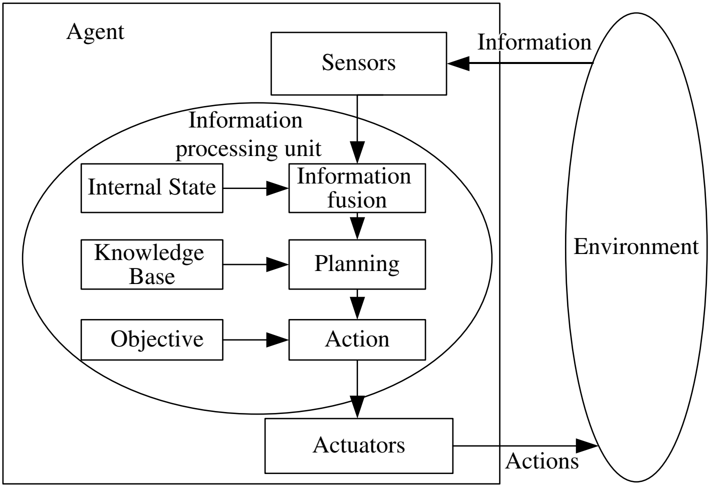

Figure 2-2 The basic structure of a thinking Agent.

The foundation of a thinking Agent is Bratman's philosophy,[5] in which rational balance between the faith, desire and intention are maintained so as to effectively solve the problem. Its information-processing unit is adopted to plan the information model of environment and internal system by knowledge base to perform logical reasoning, and to perform knowledge-based decision-making in terms of pattern matching and symbolic operational decisions. The key issue of this mechanism, which follows the classical AI, is to map a real environment to the information model.

The most important feature of a thinking Agent is to treat an Agent as a kind of intentional system, which can simulate all the intentional stances of the human individual and social behavior, for instance, belief, desire, intention, goal, commitment and responsibility. Although the concepts of intentional system and intentional stances are introduced as a natural and intuitive way to understand, describe, standardize, reason and predict the structure of an agent, operating rules and changes in status of an Agent, which help researchers inherit the results of AI, the project also faces many challenges from actual requirements.

The prime problem in constructing a thinking Agent is to choose an intentional stance. Various classifications of intentional stances can be obtained by using different Agent models or systems. Currently, the representative structure of a thinking Agent is the BDI (Belief Desire Intention) Model proposed by Rao and Gergeff,[6] which describes the structure of an Agent by using three intentional stances, belief, desire, and intention, and investigates the abstract nature and reasoning process of the BDI model by using a knowledge base. The BDI theory explains how an Agent system interacts with others to complete the mapping from input to output. According to the characteristics of applications, intentional stances can be classified into four major classes by Shoham et al.:[7] information class (describing the information of an Agent), motivation class (Agent's action selection), social class (social, moral and rational behavior related to Agent), and other classes (such as joy, fear, etc.). Intentional stances have been classified into two major classes by Wooldridge et al.:[1] information class and state class. The former one refers to the information and knowledge about an Agent, the environment and other Agents, such as faith and knowledge and so on. The latter one refers to the states that lead to the implementation of the Agent's action, such as desire, goal, intention, commitment, responsibility, ability, and so on. It is believed that a rational Agent always adopts a positive attitude based on information, for example, selecting a target and forming intention in terms of belief.

The metal states described by a thinking Agent include:

1) *Belief:* the current world situation and a certain effect achieved by using the route of a behavior are estimated by an Agent. The information model is used to describe each scene that corresponds to a possible condition of the world, which represents the world that an Agent may be in.
2) *Desire:* the preferences of an agent for the future state of the world and the route of possible behavior. An Agent may possess incompatible desires, which cannot necessarily be achieved.
3) *Target:* the achievable subset of desires is pursued by an Agent, but it doesn't promise to take concrete actions.
4) *Commitment:* conversion from the target to the action, which shows the extent of insistence on the action and reflection.
5) *Intention:* to guide and supervise the actions of an Agent.

6) *Planning:* When an Agent commits to reach a goal, intention should be regarded as a part of the planning behavior. Intentions are commonly combined to form the planning behavior in terms of specific structures.

In these mental states, the relationship between belief and intention is presented as follows:

1) *Intention-belief consistency:* an Agent should believe in the possibility of its intentions and believe that goals can be obtained in the right conditions.
2) *Intention-belief incompleteness:* a rational Agent holds incomplete belief in its intention.
3) *Side effect:* an Agent intends to do action a., and it believes if A is done, then B must be done; but it is not required to have the intention to do B.

The relationship between desire and intention is presented as follows:

1) *Internal consistency:* an Agent should avoid having conflicting intentions, but it may have conflict desires.
2) *Means-objective analysis:* an intention requests that an Agent consider asking questions in the future; whilst a wish does not request that an Agent consider asking questions in the future. nor do desires.
3) *Successfully track or not:* since intention can be regarded as the combination of desire, action and fulfilled commitment, whether the intention is successful should be tracked. It should be re-planned if it failed.

2.2.2 Reactive Agent

Since a thinking Agent depends on specific knowledge and objective optimization methods, it is not able to adapt to dynamic environment and changing situations. Some researchers thought that the knowledge base cannot objectively reflect the real world due to its simple abstraction of objective things and their behavior patterns in the real world. And they thought that the intelligent behavior of an Agent depends on perception and actions, and the overall performance of an Agent can be shown in the interaction with the real world and the surrounding environment. Therefore, an intelligent behavior structure called reactive Agent has been proposed. It has been noted that An Agent does not need knowledge, representation and reasoning – it only needs a variety of behavior patterns – and it evolves in terms of the "perception-action" mechanism. The behavior of an Agent is the result of mutual collaboration and competition of many entities that try to control the behavior of the Agent, and the basis of its behavior is this competition. In this "perception-action" mechanism, the information model of the external environment and internal state are not required to be developed during information processing. Complicated reasoning and decision-making are not required to be conducted, and the action sequences are obtained directly, which shows good robustness and fault tolerance.

The "perception-action" mappings are preset in the internal part of a reactive Agent. When certain conditions are met by the environmental information, an Agent directly calls the preset perception-action mappings to generate its corresponding action output. The basic structure of a reactive Agent is shown in Figure 2-3. In the figure, information perception and actions are combined by the perception-action relationship base, which interacts with the environment through sensors and actuators.

In contrast to a thinking Agent, a reactive Agent responds quickly; mainly because it does not contain a logical reasoning module and does not adopt a complex reasoning system.[8] Experimental results indicate that the processing speed of a reactive

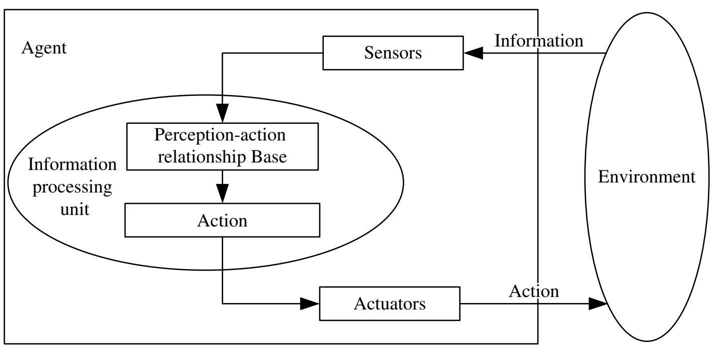

Figure 2-3 The basic structure of a reactive Agent.

Agent is faster than that of a thinking Agent when processing a limited number of tasks in the real world. However, there are some problems for a reactive Agent when it handles tasks requiring knowledge of the environment. This knowledge must be obtained from memory or by inference, not by perception of sensors. In addition, a reactive Agent has relatively poor reasoning ability, it is not able to learn, and each behavior of a reactive Agent must be encoded separately, which also leads to poor system scalability.

2.2.3 Hybrid Agent

A thinking Agent has high intelligence, but it is unable to respond quickly to changes in the environment, and its efficiency is relatively low. A reactive Agent is able to respond quickly and in a timely fashion to changes in external information and the environment, but it has low degree of intelligence, and it lacks flexibility. A reactive Agent has two major weaknesses: (1) its decision is made on the basis of local information rather than considering the whole and the remainder of the information; it cannot predict the effect of its decision on the overall behavior: that is to say, it lacks foresight, which may lead to instability of system behavior; (2) the relationship between individual behavior, the environment and the overall behavior of the system cannot be understood, which makes it very difficult to design an Agent to complete the assigned tasks; designers must go through many experiments and error-modification processes to design a better Agent.

Neither a pure thinking Agent nor a pure reactive Agent is appropriate for most of the practical problems; it is better to combine the advantages of both structures to form a hybrid Agent structure. The multi-layer structure is commonly adopted, in view of different requirements of time in response to questions; the structure of the different layers is set to be a reactive agent or a thinking Agent accordingly.

In general, the structure of a hybrid Agent is designed as a hierarchical structure that consists of at least two portions: the upper layer is a cognitive layer containing a knowledge base, which plans and conducts objective decision by using the traditional logic reasoning approach; the lower layer is a reaction layer that can respond quickly to and deal with emergencies in the environment, which doesn't adopt any symbol representation and reasoning system. In addition, the reaction layer is generally given a higher priority. There are two important issues with respect to the hierarchical structure. One issue is what kind of framework should be adopted in each layer. The other issue is how the layers should interact with each other.

A typical example of a hybrid Agent is the Procedural Reasoning System (PRS), which is a BDI system to reason and perform tasks in a dynamic environment.

Currently, the thinking Agent is the domain in which the BDI structure is most popular in the research and application areas due to its solid theoretical foundation and easy operability. Research in the area of the reactive Agent is still in its infancy. The hybrid Agent has its own characteristics: (1) the hybrid Agent is applied in different areas; (2) in the procedure of design, it is possible to apply one type of structure to another area after adjustment; (3) rather than using one type of structure to develop a Multi-Agent System (MAS), a hybrid design with integrated advantages of various types of structures should be adopted to solve many problems. Indeed, a hybrid Agent becomes a hot topic because a hybrid Agent combines the advantages of two Agents and the problem itself is diverse.

2.3 The Structure of a Multi-Agent System

2.3.1 The Environment of a Multi-Agent System

In order to solve complex problems that cannot be solved by a single Agent, a Multi-Agent System is developed by multiple Agents collaboratively to form a problem-solving environment. A Multi-Agent System is an intelligent society composed by a number of Agents, which can solve a distributed problem. These Agents analyze and reason according the messages received by mutual communication, and learn from experience.

When a Multi-Agent System is in the environment, the environment has the following typical characteristics:

1) The environment of a Multi-Agent System provides an infrastructure for communication and interaction protocols among Agents.
2) The environment of a Multi-Agent System is mostly open.

3) The environment of a Multi-Agent System consists of independent, distributed Agents, cooperative relationship among agents may exist.

2.3.2 The Structure of a Multi-Agent System

In terms of the characteristics of its environment, a Multi-Agent System designs the competition and collaborative relationship among Agents, and determines the organizational structure of a system. The organization of a Multi-Agent System determines the behavior characteristics of a system, the interactive way among Agents, and the problem-solving structure, which has a great impact on solving efficiency and system performance. The structure of the existing Multi-Agent System can be classified into three major classes: hierarchical structure, federal structure and fully autonomous structure.[10]

2.3.2.1 Hierarchical Structure

Figure 2-4 represents a MAS with a multilayer structure, and its managing style is top-down. The notable feature of a hierarchy structure is that there is information exchange among Agents in the same layer, and it is a loose "master/slave" relationship among Agents in different layers. Although Agents in the lower layer are under the control of Agents in the upper layer, it has a certain autonomy and intelligence. The control method is presented as follows: Agents in the lower layer are started up by Agents in the upper layer. In each layer, an Agent initiates the negotiation process with associated Agents in the same layer. Only when important events happen or Agents in the lower layer cannot

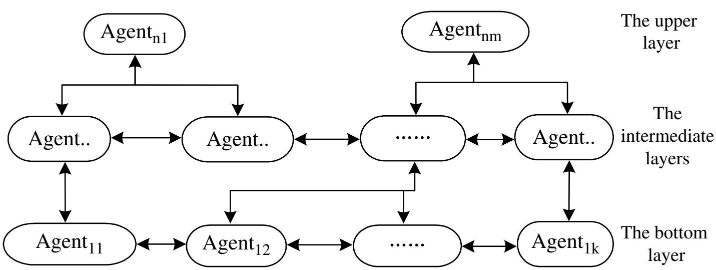

Figure 2-4 The hierarchical structure of a Multi-Agent System.

reach an agreement, which affects the realization of the overall objective, would the negotiation process be coordinated by Agents in the upper layer. The advantages of this hierarchical structure are as follows: its response time to disturbance is short; reliability and fault tolerance have been greatly improved; Agents in the lower layer can run independently when Agents in the upper layer cannot work. Examples of a hierarchical structure include hierarchical manufacturing systems,[11] multi-layer production planning and control systems,[12] and so on.

2.3.2.2 Federal Structure

The coordination mechanism based on a mediator are introduced in the federal structure. As shown in Figure 2-5, there are three mediators: each mediator gathers a group of Agents to become an Agent set, and each Agent in the Agent set coordinates communication and behavior by a mediator. Meanwhile, the mediator on behalf of the entire Agent set conducts coordination of communication and behavior with other mediators in the system.

The federal structure of a Multi-Agent System reduces the amount of communication of coordination activities among Agents in the Multi-Agent System through the mediator Agent, which ensures the stability and extensibility of the system so as to obtain a large number of applications. The federal structure

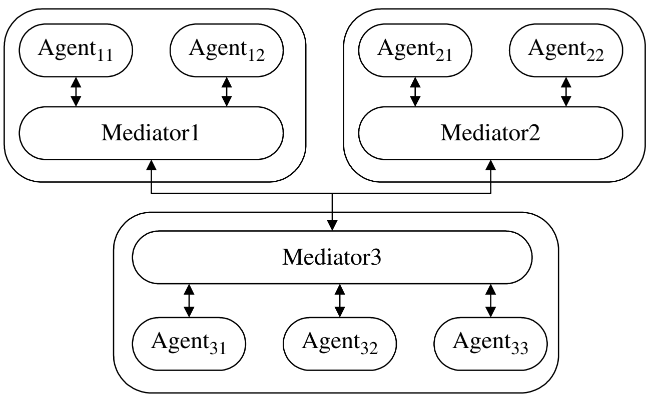

Figure 2-5 The federal structure of a Multi-Agent System.

provides an open and scalable framework for the development of a Multi-Agent System.[13] Agents in the federal structure of a Multi-Agent System undertake different tasks so as to simplify the computational complexity of the system and increase the controllability of the system. Therefore, in particular, it is more suitable for the development of a complex and dynamic intelligence system composed of a large number of Agents.

2.3.2.3 Fully Autonomous Structure

All Agents in a Multi-Agent System with fully autonomous structure are autonomous and equal; its structure is shown in Figure 2-6.

Although there are various definitions of the autonomous Agent, it is generally accepted that the autonomous Agent should have at least the following four characteristics in common: 1) The Agent cannot be controlled or managed by other Agents or people, and there is no central control Agent; 2) The Agent can communicate or interact directly with any Agent in the system and external systems, and control through coordination of Agents; 3) The Agent has knowledge of other Agents in the system and environment; 4) The Agent has its own interest targets and its corresponding motivation. Hence, the fully autonomous structure of a Multi-Agent System possesses several advantages over other structures. Firstly, it is able to track and deal with random changes and has a high system sensitivity, which can effectively improve the fault tolerance of the system.

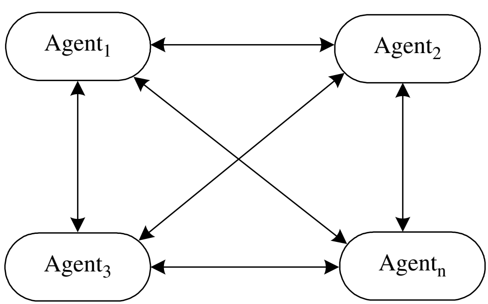

Figure 2-6 The fully autonomous structure of a Multi-Agent System.

Next, its unified system structure and mutual collaboration improve the modularity and flexibility of the system, and reduce the complexity of the system, which is good for decreasing costs of system development. For example, a distributed intelligent design system is developed by using the fully autonomous structure of a Multi-Agent System, in which intelligent design tools distributed in different locations are packed as an Agent to provide service for the system. The fully autonomous structure of a Multi-Agent System can also be applied in an autonomous multi-robot system.

When the autonomous structure of a Multi-Agent System is applied in fully distributed engineering systems, there are obvious defects as follows:

1) *It is difficult to obtain the global objective*. Each Agent is always trying to achieve its own objective regardless of the overall objective. The global optimization capability of distributed systems is regarded as a difficult problem in the area of the distributed artificial intelligence research.
2) *The instability of the system*. The interaction among Agents gives the system unstable dynamic characteristics, and thus it is difficult to predict the behavior of the system, and there is a "deadlock" and "live lock".
3) *Large amount of communication*. Agents mutually share information and knowledge. When the system is large-scale, the number of Agents in the system increases, and the amount of communication will be greatly increased.
4) *Sensitivity of coordination rules*. This structure mostly adopts a rule-based implicit coordination manner, for example, in a Multi-Agent-based manufacturing system, the market-rules-based coordination is mostly applied. In such a system, sometimes small changes in market rules will lead to major changes in system behavior.

Therefore, the fully autonomous structure of a Multi-Agent System is suitable for small-scale systems. When the system size becomes larger, with the increase in the number of Agents, the communication amount of the system will increase, and the system structure will be very complicated. Although the local autonomy of this structure is very good, it is not easy to achieve global optimization objectives.

2.4 Modeling Methods of a Multi-Agent System

Each Agent in a Multi-Agent System is physically or logically dispersed, and its behavior is autonomous. They are linked following a specific type of structure so as to complete a common task or achieve certain goals, and effects are revealed by organizational behavior. In order to describe various organizational behaviors of Agents in a Multi-Agent System, a MAS modeling method is developed in this section to describe Agents. The Unified Modeling Language (UML) is a common visual modeling language on the basis of object-oriented ideology with the goal to build object-oriented systems clearly and intuitively. The ULM modeling approach provides nine model diagrams: use case diagram, class diagram, object diagram, sequence diagram, collaboration diagram, state diagram, activity diagram, which provides support to describe the system behavior and interaction relationship from different perspectives, therefore, the MAS modeling method based on UML is presented in this section. Considering that the use case diagram is applicable to describe the object function from the user's perspective, a use case diagram is used to build the behavior model of an Agent in this book. Meanwhile, since the sequence diagram is able to describe the sequence relationships of the interaction among objects, a sequence diagram is used to construct sequence relationships of the interaction among Agents in the MAS in this book.

2.4.1 The Behavior Model of a Multi-Agent System

A use case diagram is adopted to develop the behavior model of single Agent in a MAS, and to describe the system functions and the corresponding operator, which is simple and easy to understand and communicate.

The accounts receivable and payment of manufacturers in the production management process is presented as shown in Figure 2-7. The accounts receivable process of manufacturers involves a vendor Agent, a manufacturer Agent, a bank assistant Agent, and an account Agent, and so on. Among them, the vendor Agent is responsible to issue bills for the bank assistant Agent; the manufacturer Agent instructs the bank assistant

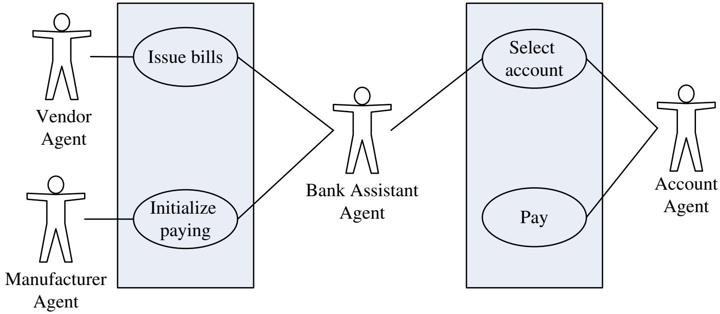

Figure 2-7 The use case diagram of accounts receivable and payment in the procurement procedure in the production management process.

Agent to pay through the payment initialization module; the bank assistant Agent selects the required account in accordance with the objective to minimize the cost of transfer; the account Agent pays the bill. The use case diagram describes each function of the system and its corresponding operating Agent during the procedure of accounts receivable and payment of procurement in production management process.

2.4.2 The Running Model of a Multi-Agent System

The sequence diagram is adopted to present the running model among Agents in a MAS, which emphasizes the sequence of information sent among Agents, and also presents the interaction among Agents. The sequence diagram has two dimensions: the vertical dimension describes the occurrence of information/call chronologically; while the horizontal dimension represents the Agent object that the information is sent to.

The accounts receivable and payment of manufacturers in the production management process is illustrated in Figure 2-8. In the accounts receivable process, firstly, the manufacturer Agent orders products through the order module from the vendor Agent. Secondly, the vendor issues bills to the bank assistant Agent. Thirdly, the manufacturer Agent instructs the bank assistant Agent to pay through the payment initialization module. Fourthly, the bank assistant Agent selects the required account in accordance with the objective of minimizing the cost of transfer. Finally, the account Agent pays the bill.

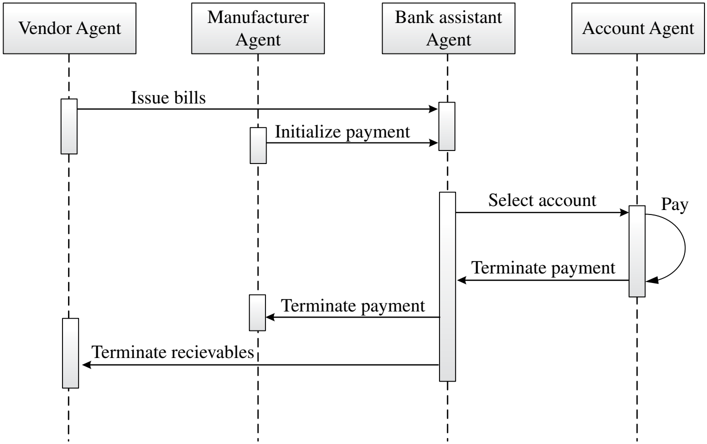

Figure 2-8 The sequence diagram of accounts receivable and payment of procurement procedure in the production management process.

In contrast to the use case diagram, the sequence diagram is used to express the function execution sequence of different Agents in the system during the receivable and payment of procurement procedure in the production management process from the perspective of function execution.

As the previous presentation shows, it is clear that the main purpose of a use case diagram is to understand the function and its corresponding role of a MAS in a visual way. Nevertheless, a sequence diagram focuses on the interaction process of each Agent in a Multi-Agent System. The UML-based MAS-integrated modeling method fully takes advantage of the model diagram in the UML, which visually represents various behaviors of a MAS by using the extended UML graphical symbols. Moreover, auxiliary maps are converted to express an Agent's characteristics, such as autonomy, reaction, and so on, and the consistency of any model at each stage during the development of a MAS. It has the following significant features:

1) *To present the characteristics of an Agent comprehensively:* the use case diagram shows its autonomous behavior, and the sequence diagram indicates the interaction between them, and so on.

2) *A systematic modeling method:* this method contains a visual model diagram, the model diagrams at various stages are compliant with the strict diagram conversion mode, and the method supports the whole process from requirements analysis to the model realization.
3) *Modular and reusable:* there are modular component design features from the requirements analysis phase to the design phase, and these reusable modules are easy to reconstruct.
4) *Strong practicability*: the use case diagram is adopted to form a document by using visual models, which helps the design developers to understand the system requirements and functionality, and also helps them to communicate with customers. Meanwhile, many UML-based modeling tools support the extension of UML, and support generating a source code framework from the model directly, which makes it easy to implement the model.

2.5 The Communication and Interaction Model of a Multi-Agent System

The behavior of communication and interaction among Agents in a MAS is an important manifestation of the sociality of an Agent, it is a way for an Agent to interact with the external environment, and it is also an important feature to distinguish a MAS from traditional AI systems. The process of communication and interaction is a relatively broad concept, which refers to interactive behavior among Agents in any form and any depth.

The mechanism of communication and interaction among Agents represents the intrinsic factor of the interactive behavior among Agents. It is not only the basis of collaboration among Agents, but also a premise to reflect the organizational relationship among Agents in a MAS.[14] Three problems are solved by this mechanism: Why interact? Interact with whom? How to interact? The MAS runs in an environment in which each Agent can efficiently run, and all Agents can interact with each other: the environment provides the infrastructure for such interaction. The basic framework must be hierarchical according to the inherent hierarchy feature of an Agent shown in the information-transfer process and the hierarchical structure of a computer network protocol. The entire communication and interaction behavior is divided into three levels: the transport layer, the communication layer, and the interaction layer. The communication and interaction behavior in the lower layer serves the behavior in the upper layer, and the communication and interaction behavior in the upper layer is implemented on the basis of the behavior in the lower layer in order to generate a hierarchical service relationship.[15]

Figure 2-9 illustrates the structure of the communication and interaction behavior between two Agents. The transport layer at the bottom (i.e., the computer network protocol layer) is a layer for the final application, which is responsible for expressing the message in the intermediate layer (i.e., the communication protocol layer) by using a specific computer network protocol in order to ensure that various interaction behaviors among Agents are realized. The network protocol is TCP/IP or HTTP, which is determined by the application of an Agent. Since the focus of this book is not on its theory and implementation, it is assumed that the interaction in this layer is achieved in the following discussion.

The layer in the middle is the communication layer (i.e., the communication protocol layer), which is mainly used to ensure that Agents CAN exchange messages with each other and understand the information. The messages mostly have clear intentions such as instructions, promises, proposals, rejections,

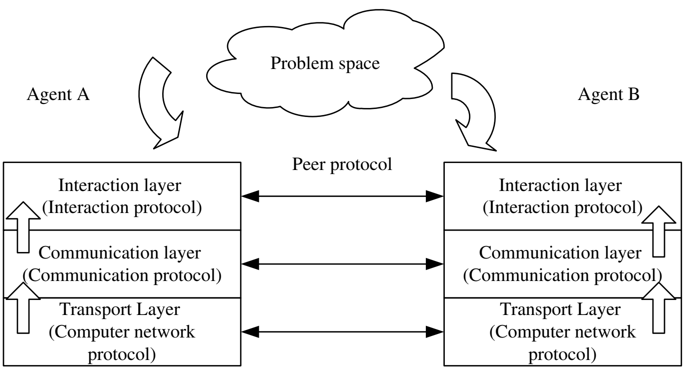

Figure 2-9 The hierarchical model of the communication and interaction process among Agents.

and so on, which reflects the initiative of an Agent as a conscious system. In the current study, the communication protocol layer is generally developed on the basis of a speech act theory.

The upper layer is the interaction layer (i.e., the interaction protocol layer). Its role is to make sure that Agents can exchange messages containing a certain structure with each other. If the aim of the communication protocol layer is to make the interactive sides understand every word of each other, then the purpose of the interaction protocol layer is to make the interactive sides collaborate by using a series of dialogues under the guidance of strategies from the upper layer. The interaction protocol layer is one important issue with respect to the interaction mechanism of Agents. There are many successful interaction protocols such as blackboard structure, contract net protocol, voting protocols, and auction protocols, and so on.[16]

Communication languages and interaction protocols in the communication layer and the interaction layer of the MAS communication and interaction hierarchical model will be presented in the following section.

2.6 The Communication Protocol for a Multi-Agent System

The communication protocol of a MAS is a convention of communication between two sides, the current MAS communication protocol is mainly based on the principle of speech act theory, which provides the basic idea of MAS communication, and ensures mutual understanding and message exchanges between Agents. In terms of the speech act theory, communication is treated as an action, which is based on the assumption that verbal action performed by an Agent is to promote its intention as to other actions. According to the research achievements of the speech act theory, the communication protocol between Agents is expressed by the communication language and the communication ontology. The communication language between Agents should be able to present the content and intent of a message with a concise syntax and clear semantics, and the background knowledge of the message should also be mastered by all communication parties so as to ensure high efficiency and clarity of delivery messages. The communication ontology is adopted to represent the basic terminology used in this field. There are numerous communication languages for an Agent that have been developed to date, for example, KIF (Knowledge Interchange Format), KQML (Knowledge Query Manipulation Language) developed by DARPA (the Defense Advanced Research Projects Agency), and FIPA-ACL (FIPA: Foundation for Intelligent Physical Agent, ACL: Agent Communication Language).

2.6.1 Communication Languages for an Agent

2.6.1.1 KIF

The KIF language was originally developed as a common language to express the features in a specific area. Instead of defining the message itself, it is a language to represent the content of a message. KIF is strictly based on first-order logic. For example, an Agent can be expressed by using KIF as follows:

1) Properties of things in a field (e.g., equipment manufacturing enterprises belong to the discrete manufacturing industry – equipment manufacturers have attributes of the discrete manufacturing industry).
2) The relationship between things in one area (e.g., when assembling computer accessories and a chassis shell to form a computer, there is an assembly relationship between computer accessories and chassis shell).
3) General properties of a field (e.g., every production process takes time).

In order to express these things by using KIF., assuming that a basic fixed logical structure consists of common first-order logic connectors, binary Boole variable connectors *and*, *or*, *not*, and so on, the universal quantifier *forall* and existential quantifiers *exists*. Furthermore, KIF develops a basic object vocabulary, in particular, numbers, characters, strings, and also develops a number of standard functions, and the relationship between objects such as the less-than relationship between numbers, the addition function, and so on. Symbols similar to LISP language are also developed to represent objects. The objects, functions, and a new relationship between these objects can be defined by using this basic structure.

2.6.1.2 KQML

KQML[17, 18] is a relatively successful Communication Language for an Agent, and its ideology is developed on the basis of a speech act theory. KQML was initially proposed in the knowledge-sharing plan supported by U.S. Defense Advanced Research Projects Agency, which defines the smallest unit in the process of information interaction-message structure, syntax and semantics, and so on, and its characteristic is that all message content is contained in the communication information.

KQML is an Agent communication language based on messages; KQML has defined a public format for information. A messages in KQML is an object: each message has a predicate (i.e., class of messages), and multiple parameters (i.e., attributes and attribute values, which are instantiated variables). In the 1990s, several different versions of KQML appeared, and each version has different collection of predicates. Finin[18] et al. summarized predicates in each version to generate a new version which contains 41 predicates.

Here is an example of a KQML message:

(ask-one
:content (PRICE IBM server? price)
:receiver server price
:language LPROLOG
:Ontology NYSE-TICKS

)

The intuitive explanation of this message is that the sender is asking the price of IBM servers. The predicate is *ask-one,* that is to say, one Agent asks another Agent, and an answer should be given. Other parts of this message represent its properties. The "*content*" domain is the most important attribute, which defines the content of the message. In this case, the content is only to inquire the price of IBM servers. The "*: Receiver*" attribute represents expected recipients of messages; the "*: language*" attribute specifies that the expression content is a language called LPROLOG (assuming the recipient "understand" LPROLOG); "*: Ontology* " (Ontology) attribute defines a term used in the message.

2.6.1.3 FIPA ACL

In 1995, FIPA began to be engaged in defining the system standard for an Agent. The core of the first step is to develop ACL.[19] FIPA ACL is similar to KQML, and it defines an "outer" language for messages. It has 20 kinds of predicates (e.g., *inform*) to define the expected interpretation for messages. It does not specify any particular language for the message content. In addition, the specific syntax of FIPA ACL messages is very similar to that of KQML. Here is an example of FIPA ACL messages:

(inform
:sender agent1
:receiver agent2
:content(price good2 150)
:language sl
:ontology hpl—auction

)

In this example, it is obvious that FIPA communication language is similar to KQML: the structure of a message is the same, and the attribute domain of a message is also very similar. Relationship between FIPA ACL and KQML were discussed in FIPA.[19]

2.6.2 The Communication Ontology for an Agent

Ontology is a formal definition of knowledge. The general ontology to construct an Agent includes a structure, a basic classification, a relationship between subsets, and a definition of the relationship between related things. If two Agents communicate on issues of a field, then they must agree on the terms to express this area. For example, an Agent is buying goods (e.g., nut or bolt) from another Agent, and the purchaser must be able to tell the seller the features of the items needed (e.g., size). Therefore, Agents must be able to agree on the meaning of "size", "inch", or "cm", and other terms.

FIPA defines the standard ontology for an Agent communication, which is required by an Agent communication, it consists of object description ontology, function description ontology, and exception description ontology.[19]

1) Object description ontology consists of objects involved in the dialogue of the implementation of an Agent management

process, which includes Agent identifier description, service description, search constraint description, property description, and so on.
2) Function description ontology consists of operations involved in the implementation of an Agent management process, which includes Agent registration, Agent cancellation, Agent modification, and so on.
3) Exception description ontology consists of exceptions that occur in the implementation of an Agent management process, which mainly includes incomprehension exception, refuse exception, and failure exception.

2.7 The Interaction Protocol for a Multi-Agent System

The MAS communication protocol ensures that the communication parties can exchange and understand messages; a series of organized and structured message is exchanged to express the intent of an Agent in a MAS. This organized and structured message exchange process, driven by certain intent, is regarded as a dialogue process. The interaction protocol is the abstract and regulation of a relatively stable dialogue process.

The interaction protocol for a MAS directly reflects the purposes and rules of interaction between Agents, which is closely related to the internal information-processing mechanism of an Agent. Therefore, the focus of a MAS is on the interaction protocol.

2.7.1 Classification of Interaction Protocols

Interaction protocols can be classified according to various criteria. In general, there are two classification methods: those based on the interaction purpose and those based on the action time of the interaction protocol.

The method based on the interaction purpose is to consider the tightness of the relationship between Agents. This can be subclassified into methods based on collaboration and those based on negotiation, which are referred to as "collaboration protocol" and "negotiation protocol". "Collaboration protocol" emphasizes that interaction parties have consistent, or temporarily consistent interests, and the purpose of collaboration is to help them cooperate with each other in order to achieve a common goal: for example, the contract net and the blackboard structure are commonly used to solve collaborative problems. Unlike the "collaboration protocol", the "negotiation protocol" is a competitive or self-serving interaction protocol. The aim of an Agent using this interaction protocol is to maximize its own interests, for instance, voting mechanism, auction mechanism, negotiation mechanism, debate mechanism, and so on.

In terms of the action time, interaction protocols can be further classified into three classes: the long-term protocol, the mid-term protocol, and the short-term protocol. The long-term protocol provides interaction rules for an Agent in a very long time, for example, in the MAS organization, the long-time commitment of an Agent limits the interaction pattern and content of an Agent so as to generate a long-term interaction protocol between roles. When an Agent plays its corresponding role, its interactive activities must be carried out in accordance with the requirements of the "role", and its corresponding commitments of the "role" should be fulfilled. On the other hand, the short-term protocol gives interaction rules for each Agent in a specific task, and may even be one-time, for instance, a contract net protocol, in which an Agent obeys the protocol regulations only on a specific task; the protocol is terminated when the task is completed. The action time of the mid-term protocol is between the former two protocols. In general, it is an interaction protocol based on planning, for example, local wide-area planning; it emphasizes that the activity planning process between Agents in the future should be negotiated interactively to generate a wide-area planning process involving each party (or contracts, protocols), and the local planning process should be modified in accordance with the wide-area planning process so as to guide their future behaviors in a certain period.[21, 22]

In fact, there is no strict definition in the classification of interaction protocols. Some comprehensive interaction protocols may be adopted by an Agent or a designer for a specific reason. Its purpose is to ensure the organization of the dialogue process so as to reflect the intent of the interactive parties and achieve a certain objective.

2.7.2 Description of Interaction Protocols

In terms of the purpose and process of interaction protocols, an overall description of various types of interaction protocols are given in this section.

Let A denote the set of all Agents; S denotes all the possible states of an Agent; B denotes all the possible interactive behaviors of an Agent. If the power set of a set Q is $\wp(Q)$, then a communication protocol InP can be defined as a six-tuple.

$$InP = \left\langle Ag, St; Ob, R, Bh, Act \right\rangle$$

Where $Ag \in \wp(A)$ denotes objects involved in the interaction protocol. It consists of Agents related to interest: for example, the auctioneer and bidder in a contract net protocol, the parties of interest conflicts and the arbitrator in the negotiation protocol, and so on.

$Sg \in \wp(\mathrm{S})$, denotes each interactive state during the interaction process of an Agent. A state is always related to a specific interaction between Agents. For example, in the contract net protocol, it is a state after the auctioneer issues the bidding information for potential bidders and another state after receiving responsive bidding information, and the target state is reached when solving the problem.

$Ob \in \wp(S)$, and $Ob \subseteq St$, denotes the target state that the interaction protocol is to reach. Target state reflects the rational pursuit of an Agent, for instance, some evaluation indexes. In addition, the target can be clearly pregiven, and it can also be determined in the interactive process.

$R \subseteq St \times St$, denotes the direct relationship between interactive states. The process from the initial state to the target state is clarified by defining the state relationship and combining behaviors of Agents.

$Bh \subseteq \wp(\mathrm{B})$, denotes the interactive behavior allowed in the interactive process of an Agent. It is subclassified as passive behavior and active behavior. Passive behavior describes the ability of an Agent to receive actions implemented by other Agents, namely capability of sensing-related events, for example, in the contract net protocol, the capability of the auctioneer and the bidder to mutually receive bidding information. In the negotiation protocol, the ability of a party to receive proposal, approve and oppose. Proactive behavior describes the ability of an Agent to implement behavior initiatively, for example, in the contract net protocol, the ability of the auctioneer to call for bidding, and the ability of the bidder to bid. In the negotiation protocol, the ability of a party to propose, anti-propose, approve and oppose.

$Act : R \times St \rightarrow Bh$, denotes the relationship between changeover of interaction states and the specific interactive behavior under the influence of state relationship. On the one hand, changeover of the interaction state of an Agent is always related to a certain behavior of an Agent. For example, in the contract net protocol, the status of the bidder is changing because it receives a bidding notice, issues bidding information, or completes a contract. On the other hand, any effective behavior implemented by an Agent should be associated with its corresponding interaction state based on the state relationship. For example, in the contract net protocol, under the state that the auctioneer receives the responsive bidding information issued by the potential bidders, the effective behavior of the auctioneer at this time is to evaluate and select the object, and send messages to confirm the bidder, and so on.

A general description of communication protocols is given above. In the practical application, there are two important issues to ensure the effective implementation of the protocol. One issue is the applicable environment of an interaction protocol. For example, the contract net protocol assumes that the interaction parties are cooperative rather than hostile and the network resource is sufficient; the negotiation interactive protocol based on game theory assumes that the interaction parties are perfectly rational. The other issue is to give additional rules and conventions according to the specific circumstances of the application environment. For example, in the interaction protocol of the contract net, the auctioneer sets reaction time restrictions when bidding in order to promptly obtain the bidding message, and so on.

In addition, in order to prevent interaction protocols from deadlocks and live locks in the real environment, the analysis and design process of an interaction protocol should also be verified, and its verification process can be found in the literature.[27]

2.7.3 The Collaboration-Based Interaction Protocol

The collaboration-based interaction protocol is referred to as a "cooperative protocol", which emphasizes that the interaction parties have consistent or temporary consistent interests and cooperate with each other in order to achieve a common goal.

The main function of the collaboration protocol is to improve the performance of the system. When Agents collaborate with each other, Agents with different intentions arrange goals and resources reasonably by coordinating their actions in order to achieve individual and group objectives. If there is collaboration, multiple Agents try their best to complete a common objective. More Agents can collaborate together to complete the task so as to increase productivity. Therefore, the productivity of a group generated by collaboration is generally higher than that of a group without collaboration.

The collaboration protocol usually has the following characteristics:

Concentration: The collaboration protocol is determined by a core decision-maker, and it is a master-slave relationship between the core decision-maker and multiple participants.

Efficiency: In the implementation procedure of a collaboration protocol, all the participants work for the common objective, which ensures high execution efficiency.

Stability: No Agent has a motive to deviate from the agreed protocol.

Effectiveness: In achieving agreement among Agents, no resources are wasted.

Concordance: No dictatorship or prejudice on an Agent due to inappropriate reasons.

Common collaboration protocols include contract net, blackboard system, and so on.

1) The contract net protocol

   When the contract net interaction protocol works, the task is divided into a series of subtasks. The manager provides the contract of a subtask by using a message structure defined by the contract net protocol. The bidding notice is open to all Agents. The manager gets access to the specific knowledge and solving methods to select the most appropriate Agent and allocates the contract-related subtask to it, and the manager accepts the contract in a specific format. When the task is solved, the solved task will be sent to the manager. If the scheduled task is over the capacity and resources of a junction point, it can be further divided into subtasks, and the subcontracts can be assigned to other nodes.

2) The blackboard system interaction protocol.[29]
When the blackboard system interaction protocol works, distributed problems are solved with the support of an appropriate structure. When the blackboard system is used by a MAS to write, the blackboard is regarded as a public work area, on which an Agent can publish and obtain information, data and knowledge. When the Agent makes the corresponding decision and changes some parts, its corresponding information items can be written on the blackboard. The collaborative work items are allocated to other Agents, other Agents can obtain information through the blackboard and perform its corresponding work items. After all the works are completed, Agents will feedback results on the blackboard.

2.7.4 The Negotiation-Based Interaction Protocol

The interaction protocol based on negotiation is referred to as the "negotiation protocol", which is a competitive or self-serving interaction protocol, and the purpose of an Agent using this protocol is to maximize its own interests. When the target of each Agent in a MAS is inconsistent, the means adopted by them is to negotiate. Two or more Agents reach a joint decision in the negotiation process, and each Agent tries its best to achieve a specific objective. Agents exchange their positions with each other – there are some conflicts – then they achieve unity through concessions or other alternative methods.

The negotiation protocol is the key step for a MAS to realize collaboration, conflict resolution, and conflict handling. Due to the autonomy of an Agent, there is always a conflict between the Agents; the conflict comes from occupying limited resources. For example, two Agents are required to use a resource that cannot be shared. Since resources are limited in most cases, it is necessary to adopt the conflict resolution technique to deal with these conflicts. Neither complete conflict situations nor complete cooperation situations exist; in most cases both conflict and collaboration co-exist. Moreover, a complex system is generally developed on the basis of a mix of collaborative environment and conflict environment. Therefore, it is necessary to improve the performance of a system by the negotiation process, and to select the resource of its corresponding Agent in the conflict environment by the negotiation protocol.

Apart from stability, effectiveness and concordance, the following characteristics should be taken into consideration in an ideal negotiation protocol:

Simplicity: the negotiation protocol will reduce the requirements of an enterprise on computation and frequency width.
Distribution: a central decision maker is not required in a negotiation protocol.

Common negotiation protocols include the voting interaction protocol, the auction interaction protocol, the debate interaction protocols, and so on.

1) *Voting interaction protocol.* By specifying the "ballot" format in detail and the computing method to calculate the voting results, the voting interaction protocol mechanism supports the negotiation process among Agents on a particular issue in order to optimize the ultimate goal. In the study of human communication, Grosz et al.[29] proposed a formal model to support the voting interaction protocol.
2) *Auction interaction protocol.* It is an application of one trading mechanism in business, also known as a bidding auction, in which the sellers sell the goods to people who meet the requirements of a bid. Wellman et al.[30] used the market-oriented auction method to design the negotiation process between Agents and proposed the "market-oriented auction" negotiated interaction protocol based on the general equilibrium theory.
3) *Debate interaction protocol.* When an Agent faces a proposal or debate sent from an opponent, it should determine whether to accept or reject the proposal and debate by analyzing the conflict, and generate a corresponding debate or proposal on this basis. Sycara[31] studied the interaction of

non-cooperative multiple Agents with the background of labor negotiation, and proposed a “persuasive debate” model based on the solid reasoning and the multi-attribute utility optimization theory.

2.8 Conclusion

An Agent is an abstract entity that has independent functional characteristics of autonomy, self-adaptability, self-learning and mobility, and so on. It can act on its own and the surrounding environment, and can respond to the surrounding environment. Meanwhile, an Agent has knowledge and objectives, can communicate and interact with other Agents, and also has the ability to solve problems. This chapter starts from the micro level, various different extensions of Agent structures (such as thinking Agent, reaction Agent and hybrid Agent) are introduced in detail by analyzing the basic structure of an Agent. Moreover, the environmental characteristics of a MAS is comprehensively analyzed from the macro level, and MAS structures, modeling methods, and communication and communication protocols are discussed in detail, which provides a foundation for developing a MAS system, collaborating among Agents, and solving the planning and control problems in a manufacturing production system in the following chapters.

References

[1] Wooldridge, M., and Jennings, N. Intelligent Agents: Theory and Practice. *The Knowledge Engineering Review*, 1995, 10(2): 115–152.

[2] Jennings, N. R. Agent-based Business Process Man Agent. *International Journal of Cooperative Information System*, 1996, 5(2–3): 105–130.

[3] Jennings, N. R., Faratin, P., Norman, T. J., et al. Autonomous Agents for Business Process Management. *International Journal of Applied Artificial Intelligence*, 2000, 14(2): 145–189.

[4] Sycara, K. P., and Wooldridg, E. M. *Agents'98: Proceedings of the Second International Conference on Autonomous Agent*. New York: ACM Press, 1998.

[5] Bratman, M. E. *Intention, Plans and Practical Reason*. Cambridge: Harvard University Press, 1987.

[6] Rao, A. S., and Georgeff, M. P. Formal Models and Decision Procedures for Multi-Agent Systems. *Technical Note from Australian AI Institute in Australia*, 1995.

[7] Shoham, Y. Agent Oriented Programming. *Artificial Intelligence*, 1993, 60(1): 51–92.

[8] Brooks, R. A. Elephants Don't Play Chess. In *Designing Autonomous Agents* (ed. P. Maes), Cambridge: Harvard University Press, 1991.

[9] Ingrand, F. F., Georgeff, M. P., and Rao, S. An Architecture for Real-Time Reasoning and System Control. *IEEE Expert*, 1992, 7(6): 33–34.

[10] Genesereth, M. R., and Ketchet, S. P. Software Agents. *Communication of the ACM*, 1993, 37(7): 48–53.

[11] Van-Leeuwen, E. H., and Norrie, D. M. Intelligent Manufacturing: Holons and Holarehies. *Manufacturing Engineer*, 1997, 76(2): 86–88.

[12] Fischer, K. The Design of an Intelligent Manufacturing System. In *Proceedings of the 2nd International Working Conference on Cooperating Knowledge-based Systems*. University of Keele, 1994: 83–99.

[13] Genesereth, M., and Ketehpel, S. Software agents. *Communications of the ACM*, 1994, 37(7): 48–53.

[14] Huhns, M. N., and Stephens, L. M. Multiagent Systems and Society of Agents. In *MultiAgent Systems*. G. Weiss, ed. Cambridge, MA: The MIT Press, 1999.

[15] Gao Bo, Fei Qi, and Chen Guangxue. The Hierarchical Model on Agent Interaction. *Computer Science*, 2001, 28(8): 105–109.

[16] Gao Bo. Organizational Structure and Application of Multi-Agent System. Doctoral Dissertation, Huazhong University of Science and Technology, Wuhan, 2002.

[17] Labrou, Y., and Finin, T. A Semantics Approach for KQML—a General Purpose Communication Language for Software Agents. In *Third International Conference on Information and Knowledge Management*, 1984, 55–58.

[18] Finin, T., et al. Specification of the KQML Agent Communication Language. *DARPA Knowledge Sharing Initiative External Interfaces Working Group*, 1993.

[19] FIPA. Specification Part 2 – Agent Communication Language. The Text Refers to the Specification, 1999.

[20] Bourne, R. A., Excelente-Toledo, C. B., and Jennings, N. R. Run-time Selection of Coordination Mechanisms in Multi-Agent Systems. *Proceedings of 14th European Conference on Artificial Intelligence (ECAI)*. Berlin, Germany, 2000, 348–352.

[21] Durfee, E. H., and Lesser, V. R. Partial Global Planning: A Coordination Framework for Distributed Hypothesis Formation. *IEEE Transactions on System, Man, and Cybernetics*, 1991, 21(5): 1167–1182.

[22] Decker, K. S., and Lesser, V. R. Generalizing the Partial Global Planning Algorithm. *International Journal of Intelligent and Cooperation Systems*, 1992, 1(2): 319–346.

[23] Beer, M., d'Inverno, M., Luck, M., et al. Negotiation in Multi-agent Systems. *Knowledge Engineering Review*, 1999, 14(3): 285–289.

[24] d'Inverno, M., and Luck, M. Formalizing the Contract Net as a Goal-directed System. In *Agents Breaking Away: Proceedings of the Seventh European Workshop on Modeling Autonomous Agents in a Multi-Agent World*. V. D. Velde, Perram eds. Lecture Notes in Artificial Intelligence. Springer-Verlag, 1996, 10(38): 72–85.

[25] Jennings, N. R., Parsons, S., Sierra, C., et al. Automated Negotiation. In *Proceedings of the 5th International Conference on the Practical Application of Intelligent Agents and Multi-Agent Systems (PAAM-2000)*, Manchester, UK., 2000: 23–30.

[26] d'Inverno, M., Kinny, D., and Luck, M. Interaction Protocols in Agents. In *Proceedings of the 3rd International Conference on Multi-Agent Systems*. IEEE Press, 1998.

[27] Lacey, H., and DeLoach, S. A. Automatic Verification of Multi-agent Conversations. In *8th Annual Midwest Artificial Intelligence and Cognitive Science Conference*, University of Arkansas. Fayetteville: AAAI Press, April 2000, 93–100.

[28] Zhao Fumin. Networked Manufacturing System Based on Multi-Agent. Doctoral Dissertation, Hefei University of Technology, Hefei, 2000.

[29] Grosz, B., and Sinder, C. L. Plans for Discourse. In *Intention in Communication* (eds. P. R. Cohen, J. Morgan and M. E. Pollack). Cambridge: Harvard University Press, 1990.

[30] Wellman, M. P. A Market-Oriented Programming Environment and Its Applications Multi-Commodity Flow Problems. *Journal of AI Research*, 1993, 1: 1–22.

[31] Sycara, K. P. Persuasive Argumentation in Negotiation. *Theory and Decision*, 1990, 28(3): 203–242.

3

Multi-Agent-Based Production Planning and Control

3.1 Introduction

Nowadays, the structure of a manufacturing system trends to be a distributed structure, and its internal correlation and uncertain factors increase, which makes the manufacturing system become more and more complex. Traditional production planning and control systems can no longer satisfy the requirements. A Multi-Agent System (MAS) is a method that accomplishes tasks by communication and collaboration amongst Agents. It not only has features of a distributed system such as resource sharing, good extensibility, reliability, and real-time responsibility, but it also can solve large-scale and complex problems by collaboration amongst Agents, which give the system strong robustness, reliability and self-organization ability. Therefore, the framework of a Multi-Agent-based production planning and control system is developed in this chapter by analyzing requirements of the current manufacturing system. An intelligent production planning and control system is generated by cooperation amongst multiple Agents so as to implement distributed and flexible production and to respond rapidly to customers' demands.

3.2 Manufacturing Systems

3.2.1 Concept

3.2.1.1 Manufacture

Manufacture is a process that transforms raw material to expected products.[1]

Moreover, manufacture can be defined in a narrow way or a broad one.[2] *Special manufacture* refers to traditional mechanical manufacture, which mainly emphasizes processing and assembly. The generalized definition given by International Production Engineering Research Association is this: Manufacture refers to all the related activities in a product life time, such as product design, material selection, production planning, production process, quality assurance, operation management, market sales and service.[3]

3.2.1.2 System

The definition of a system given by Cihai is "a self-contained organization that integrates a set of same or similar items by certain order or internal relationship".[4]

The eminent scientist Hsue-shen Tsien defines a system as "an organization that is composed of several parts that interact with and depend on each other with specific function".[5]

The definition in Webster's dictionary is "an integrated whole formed by a set of interactive or mutually related factors in a certain way".[6]

In the Japanese Industrial Standards, a system is an aggregation of several organization factors that move to a common destination in a certain order.[7]

Hence, a system is an organization which includes several parts that interact and correlate with each other. A system emphasizes correlation and collaboration of components, which provides the possibility to thoroughly analyze and consolidate all the different parts.

3.2.1.3 Manufacturing Systems

The International Production Engineering Research Association issued the definition of a manufacturing system in 1990:[8] a manufacturing system is an integration of production in the manufacturing industry. In the electrical mechanical industry, a manufacturing system has the functions of design, production, delivery and sales.

Professor Chryssolouris[9] at MIT proposed that a manufacturing system was a combination of human, machine, material flow, and information flow.

In this book, the definition of a manufacturing system is given in a narrow way: a manufacturing system includes all the processes from raw material to production, and is a combination of human, machine, material, energy, auxiliary equipment, material flow and information flow. Manufacturing systems have become a hot research topic. In particular, in the past 30 years, various advanced manufacturing systems have been proposed, for instance, the agile intelligent manufacturing system,[10] the all-round manufacturing system,[10, 11] the biological manufacturing system,[12, 13] the random manufacturing system[14] and the re-entrant manufacturing system.[15]

3.2.2 Classification

Manufacturing systems can be classified in many ways, which generate various kinds of manufacturing systems. According to the manufacturing process, manufacturing systems can be classified into three categories: discrete manufacturing systems, continuous manufacturing systems, and hybrid manufacturing systems.

1) A discrete manufacturing system is a manufacturing system in which products are produced by physical format change of raw materials or assembly. The manufacturing process of this system is a complex combination of production processes of different components in parallel or series. A product contains multiple components, which have a relatively fixed product structure and a fitting relationship. A discrete manufacturing system is applied over a wide area of applications, which include electronic component manufacturing, automotive, furniture, hardware, medical equipment, and so on.
2) A continuous manufacturing system is a manufacturing system in which products are produced by chemical reaction of raw materials. In the system, materials continuously go through the same path, and valuable products are produced

by processes such as mixing, separation, molding and chemical reaction. Products produced in this system always have a V-shape product structure, which includes co-products, by-products, and large amounts of recycled products and waste. All the materials and costs have their own special processing processes. A continuous manufacturing system is applied mainly in the field of petroleum and metallurgy.

3) A hybrid manufacturing system is a manufacturing system in which products are produced by physical and chemical reactions of raw materials. This system consists of features of a continuous manufacturing system and a discrete manufacturing system. In general, the production process of a product in the system should go through continuous manufacturing processes as well as discrete manufacturing processes. In particular, it is possible that the production process may not belong to either of them, just a semi-discrete or semi-continuous process. A hybrid manufacturing system is applied in the field of cosmetics, liquor, food, drinks.

This book mainly focuses on discrete manufacturing systems, which can be classified into two categories according to the resource location.

1) *Centralized manufacturing systems*. A manufacturing system, in which resources have a centralized location, and all the production processes are performed in one plant.
2) *Distributed manufacturing systems*. A manufacturing system, in which resources have a geographical distributed location, and production processes are accomplished in multiple plants located in different countries or areas. Production processes include multiple processing phases that are composed of a series of intermediate stages. Every processing phase contains one plant or several plants. If a task can be accomplished by more than one plant, it is necessary to assign the task among multiple plants by using a certain allocation strategy. If there is a supply-and-demand relationship between two plants, then it is necessary to match time and quantities of components between them according to the product structure. The structure of a manufacturing system tends to be a distributed structure. As the internal correlation and uncertainties in a manufacturing system increase,

the production planning and control process in a distributed manufacturing system becomes more and more complex.

Discrete manufacturing systems can further be classified into three classes according to the material handling operation mode, that is Job Shop manufacturing systems, Flow Shop manufacturing systems, and Re-entrant manufacturing systems.

3.2.2.1 Job Shop Manufacturing Systems

In a large product variation and small batch manufacturing company, major components are manufactured by using a Job Shop manufacturing system. The system consists of a number of machines, and implements material movements between machines by using material-handling equipment or manpower. Figure 3-1 shows the material handling process of a Job Shop manufacturing system. Its characteristics are presented as follows.

1) *Versatile machine.* In a Job Shop manufacturing system, a machine may either have a single function or multiple functions. It is a flexible machine.
2) *Flexible material handling path.* In a Job Shop manufacturing system, material moves between various machines; different processes of a product may be produced by different machines; each machine can be automatically adjusted to effectively produce different products within a certain range; and the production sequence of components can be changed timely to meet market demands.

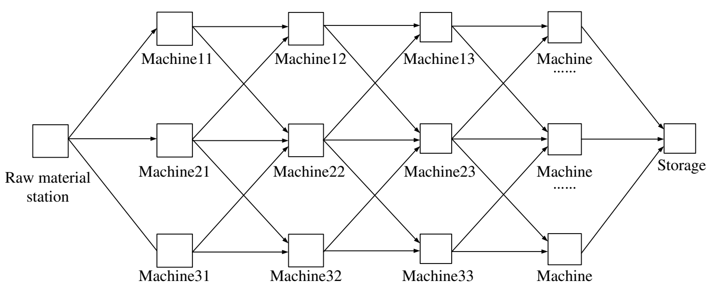

Figure 3-1 The material handling process of a Job Shop manufacturing system.

3) *Suitable for large product variation and small batch production process*. While improving the flexibility of a manufacturing process and efficiency of production simultaneously, a Job Shop manufacturing system can guarantee product quality, meet customers' demand for various functions and quickly updating. Therefore, it is very suitable for large product variation and small batch production process.

3.2.2.2 Flow Shop Manufacturing Systems

In a Flow Shop manufacturing system, material-handling equipment and machines are fixed. Figure 3-2 shows the material handling process of a Flow Shop manufacturing system.

1) *Single-function machine*. In a Flow Shop manufacturing system, machines are vertically integrated with a single function; and division of labor is very small. Such a manufacturing system has a lot of advantages such as high efficiency, high machine utilization, and low production costs. However, its disadvantages are low responsibility to market and customer demands, high one-time investment, and inflexible machines.
2) *Consistent material handling path*. In a Flow Shop manufacturing system, the production processes of different products are all the same and seldom changed. Therefore, the material handling path in such a system is consistent.
3) *Suitable for mass production process*. A Flow Shop manufacturing system is suited to manufacture similar products. If a

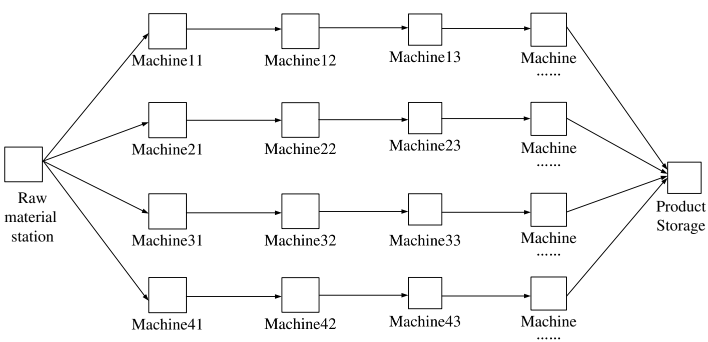

Figure 3-2 The material handling process of a Flow Shop manufacturing system.

product type is changed, it will become a costly and time-consuming job. This system is mainly applied in mass production process such as automobile manufacturing.

3.2.2.3 Re-Entrant Manufacturing Systems

When Kumar studied semiconductor manufacturing systems in 1993, he proposed a manufacturing concept called Re-entrant Manufacturing Systems (RMSs), in which the production process of wafers presents a regular re-entrant characteristic. It is completely different from that of traditional Job Shop and Flow Shop manufacturing systems. The system is developed on the basis of batch-processing machines, and single-processing machines, materials are moved by using an Automated Guided Vehicle (AGV). Figure 3-3 shows the material handling process of a RMS.

1) *Various machines*. In a RMS, batch-processing machines and single-processing machines coexist.
2) *Re-entrant material handling path*. There are many re-entrant flows in a RMS. Kumar[16] defined material handling process in a RMS as follows: wafers at different processing stages repeatedly visit the same processing station.
3) *Suitable for semiconductor manufacturing process*. Since the semiconductor manufacturing process has characteristics such as re-entrant flow, coexistence of batch-processing machines and single-processing machines, and unbalance of machine workload, which are different from those of a traditional manufacturing system, it is a typical RMS.

This book mainly focuses on production planning and control process in Job Shop manufacturing systems and RMSs.

3.3 Production Planning and Control

3.3.1 Production Planning and Control Activities

From a broad perspective, the production planning and control process of a manufacturing system involves a wide range of scopes, which includes various departments and functions within a company: for instance, product development, supply, distribution, personnel, equipment maintenance, power, customers,

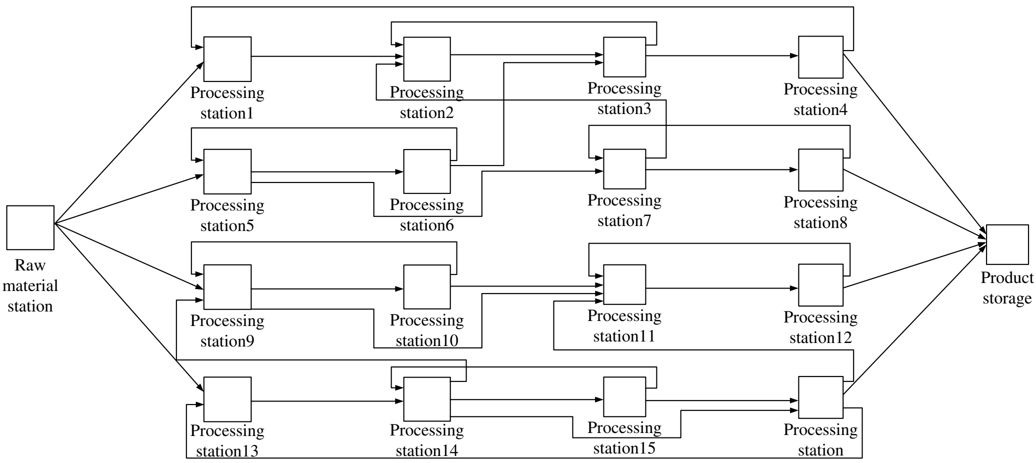

Figure 3-3 The material handling process of a RMS.

dealers, joint ventures outside a company, and so on. Top managers, middle managers and grassroots managers are involved in the production planning and control process in various degrees.[17] From a narrow perspective, the production planning and control process determines what to produce, how many to produce, and allocates resource and time for these tasks.[18]

As the structure of a manufacturing system gradually tends to be a distributed structure, an advanced and efficient production planning and control strategy indicates an increasingly important role in the enterprise competition. It involves organization and distribution of manufacturing resources, and is directly related to both the economic efficiency and the social benefits of an enterprise. Activities involved in production planning and control process mainly include:

1) *Production planning activity*. This indicates a promise by production to demand. Its major task is to determine the number of products to be produced in the planning horizon. The production planning activity mainly considers whether manufacturing resources can satisfy customers' demand. It is a crude capacity-balancing process, in which received customer demand is divided into products-production tasks, and then these tasks are grouped so as to improve the comprehensive resource utilization of the system. The production planning activity has a long time horizon, which is usually ten days or weeks.
2) *Production scheduling activity*. This indicates organization and implementation activities of production plans, which involves the specific arrangement and resource assignment in the production process to decide the time for processing each task by using a resource. According to the production plan released by the production planning activity, the production scheduling activity guides the production control activity through the production plan. Meanwhile, it is required to respond in real time to dynamic events in a manufacturing system so as to maintain the stability of a system. The aim of this activity is to maximize resource utilization. The time horizon of the production scheduling activity is shorter than that of the production planning activity, which is a day or a week.

3) *Production control activity*. This indicates tracking and feedback activities throughout the manufacturing process. The production control activity covers a wide range of contents. According to the function of an activity, it mainly consists of two issues, task execution and production data acquisition. Data acquisition includes resource data acquisition and material tracking process; the data are preprocessed to become useful for the production planning and scheduling process. For example, in the execution process of the production schedule, Kanban and barcode are often used to collect the production data. According to production factors involved in a manufacturing system, tracking activities can be further classified into three classes: production progress tracking activity, material tracking activity, and quality retrospect activity. The aim of production control is to improve the visualization of the production control process. It manages the activities within the working shift, which are real-time.

In summary, production planning activity, production scheduling activity, and production control activity in the production planning and control process of a manufacturing system are closely related. The production planning activity provides the quantity of products that customers need and the delivery due date requirements in a certain period for the production scheduling activity. The production scheduling activity arranges the specific production activities in accordance with the results obtained by the production planning activity. Whether the scheduling result is reasonable, is related to the released tasks. Meanwhile, the execution of the production schedule will affect the development and implementation of the production plan. The production control activity feeds back the execution progress of various tasks to the production planning activity and the production scheduling activity.

3.3.2 Production Planning and Control Mode

In a complicated and volatile production environment, the production planning and control strategy determines the information interaction and organization in the implementation of the production planning, scheduling and control activity so as to affect the efficiency and flexibility of the production planning

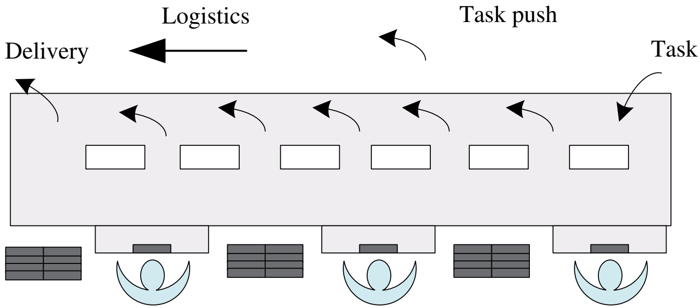

Figure 3-4 The push-based production planning and control strategy.

and to control activity to a large extend.[19] Since the 1960s, the investigation and application of the production planning and control strategy in academia and industry has undergone an evolving process. The typical production planning and control strategies include push and pull.[5, 20–25]

1) Push. This kind of production planning and control strategy is developed on the basis of future demand predicted and actual orders received. Figure 3-4 shows the task of push-based production planning and control activity.

This strategy is mainly used in a make-to-stock manufacturing system with a single product, stable demand and long delivery due date. It can improve production efficiency, strengthen system stability and shorten the product delivery due date. However, it cannot be applied in make-to-order and make-to-assembly manufacturing systems.

2) *Pull*. This kind of production planning and control strategy is to respond rapidly to customers' demand. Figure 3-5 shows the material handling pull-based production planning and control activity.

The strategy is mainly used in make-to-order and make-to-assembly manufacturing systems. With this strategy, the production execution activity is driven by the response to customer orders. It can reduce inventory, improve system flexibility and rapidly respond to customers' demand. Nevertheless, due to lack of coordination with suppliers, it will put inventory pressure onto suppliers, which cannot achieve a win-win situation.

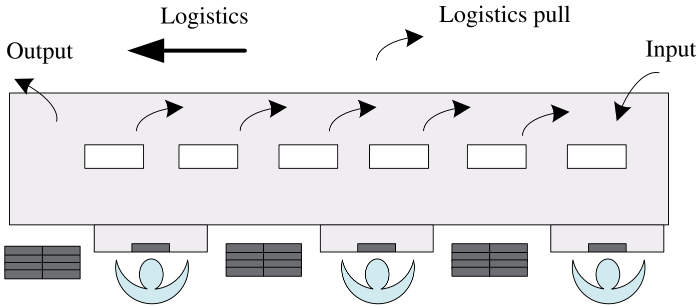

Figure 3-5 The pull-based production planning and control strategy.

3.3.3 Production Planning and Control Systems

Since the 1970s, computers have been widely applied in production management. The landmark of the production planning and control area was developed by IBM. Then the American Production and Inventory Control Society issued the Material Resource Planning (MRP) system.[26] After that, enterprises developed their production plans according to the MRP system, which included timely conducted process control, adjusted scheduling results and inventory in accordance with customers' demand and changes in the external environment.

MRP/MRP II/ERP systems are representatives of the push-based production planning and control strategy.[27] Among them, MRP is the early stage of production planning and control technology, which is developed to decide the inventory of raw materials and components production plan. MRP clearly points out that production must be oriented by market demand; the external demand (i.e., the products that will be sold as commodities) determined by the market rather than the company is regarded as "independent demand"; all the materials required for products production are regarded as "dependent demand", which describes the correlation of materials. It develops the product structure model by using time as the X-axis, and divides the final products, production materials and purchased materials into three levels. Final products, purchased parts, and processed parts are all integrated in one model so as to synchronously generate and to adjust the production plan and the procurement plan. MRP II adds the application of management accounting on the basis of MRP. And then ERP further expands the scope of information integration to the upstream and downstream of an enterprise. The developments of MRP, MRP II and ERP are treated as different levels and different solutions of enterprise applications; the latter is complementary to new demands of an enterprise. MRP/MRP II/ERP systems have a wide range of applications and affect the production management of an enterprise. With a huge and complex structure, they are a strategic business tool for an enterprise. MRP/MRP II/ERP systems are suitable for make-to-stock manufacturing systems. However, as the scale of industrial production increases, market demand diversifies, and some enterprises gradually begin to adopt the make-to-order manufacturing model so as to satisfy the customers' demand. And requirements for the production planning and control strategy are increased in the make-to-order manufacturing model.

A Just-In-Time (JIT) system is the typical representative of a pull-based production planning and control strategy, which is developed for a make-to-order manufacturing system.[28] The basic idea of the system can be summarized as "when it is needed, according to the amount needed to produce the desired product". The core of a JIT system is to pursuit either a non-inventory manufacturing system or a minimum inventory manufacturing system. Hence, a series of specific methods including Kanban are developed to generate gradually a unique production planning and control system. The successful application of a JIT system depends on four basic principles: waste elimination, participation of employees in making decisions, involvement of suppliers, and comprehensive quality management. The overall aim of a JIT system is to obtain a balanced manufacturing system, and a smooth and rapid material handling flow throughout the entire system, which consists of following sub-objectives: 1) to eliminate interruptions that are caused by low quality, machine breakdown, change of production schedule and delivery delay; 2) to be flexible, adapt to changes as variety and capacity; 3) to reduce production exchange time and production lead time; 4) to minimize inventory 5) to eliminate waste.

3.3.4 Hybrid Push-Pull Production Planning and Control System

As the structure of a manufacturing system tends to be a distributed structure, the internal correlation intensifies, dynamic uncertain factors involved in a manufacturing system increase (i.e., sudden changes in order, machine breakdown, operators' fault), the manufacturing system becomes more and more complex. The traditional push (or pull) production planning and control strategy cannot satisfy requirements of the actual manufacturing environment. The complexity of these production planning and control problems in the dynamic uncertain manufacturing environment has led to the recent interest in addressing the problems by using a hybrid push-pull production planning and control system.[29–33] The framework of a hybrid push-pull production planning and control system is presented in this section, as shown in Figure 3-6. The method is to achieve a good balance between two strategies by effectively organizing production planning, scheduling and control activities so as to meet the management requirements in the dynamic uncertain environment and to maintain the flexibility and stability of the system.[34] This system is composed of a push production planning subsystem, a hybrid push-pull production scheduling subsystem, and a pull production control subsystem.[35]

3.3.4.1 Push Production Planning Subsystem

A mathematical model is developed to describe uncertain factors incurred in the production process of a manufacturing system to implement the production planning process, which determines a reasonable time resource allocation scheme in order to formulate a flexible production plan. The central part of a push production planning subsystem is how to develop a mathematical model to express uncertain factors. A predicted production plan with certain flexibility is generated in this subsystem to support the production scheduling process.

3.3.4.2 Hybrid Push-Pull Production Scheduling Subsystem

A hybrid push-pull production scheduling subsystem is developed on the basis of a push production planning subsystem. The production schedule with anti-interference ability is generated by

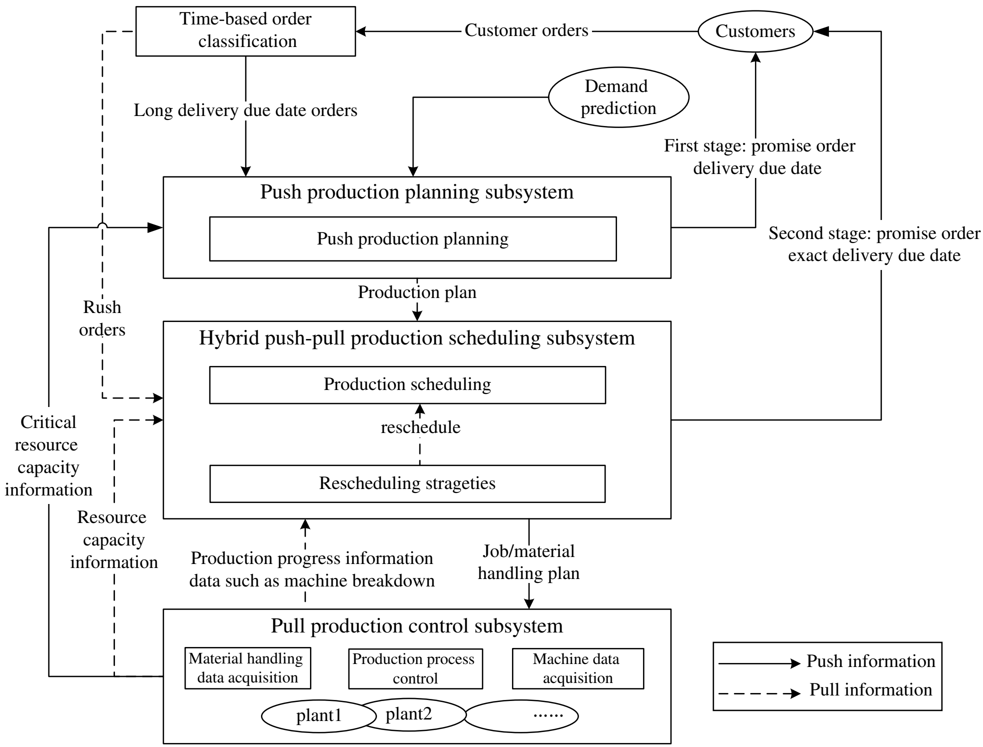

Figure 3-6 A hybrid push-pull production planning and control system.

receiving a flexible production plan. Since it cannot completely eliminate the effects of all dynamic events, it is necessary to evaluate dynamic events in the process to implement the production schedule. After the extent and scope of the effects of dynamic events are clarified, a rescheduling scheme is formulated by modifying the original scheduling scheme as little as possible to maintain the overall stability of the scheduling scheme, and eliminate the effects of dynamic events.

3.3.4.3 Pull Production Control Subsystem

Real-time production data is acquired. Since there are differences in data collection methods, data formats and analysis approaches of quality, machine, and capacity, it is necessary to study the integrated processing technology of heterogeneous data. After that, the subsystem will feed back the real-time production data and pull the rescheduling activity.

The running flow and data interaction process of a hybrid push-pull production planning and control system is very complex. The following requirements on system performance are requested by enterprises to operate the system stably.

1) *Reliability*. This is the length of time that is required to ensure that a system continuously runs without failure according to a specific indicator. Due to intensified international competition and increasingly fluctuating market demand, enterprises should decide whether to accept orders, what order to accept and when to deliver the order. Otherwise, enterprises may lose their clients or even the entire market. In the manufacturing process, a production planning and control system should adjust the original production plan in real time for coming orders, so as to respond rapidly and to ensure the stability of plans, operators and machines. Therefore, reliability is the basic principle to ensure that an enterprise can respond rapidly to customers, and maintain the stability of the production process.
2) Scalability. It is the feasibility to expand the system functions and add in new function units, and the capability to accept new features and functions. The expanding feature of a manufacturing system requests that a system can be extended according to requirements.
3) *Adaptability*. This refers to the ability that a system must adapt to the fast-changing manufacturing environment. The system is

required to adapt to unpredictable environmental changes, and to meet both various requirements of the market and the requirements of manufacturing resource distribution and function distribution in a distributed manufacturing system. Moreover, it should support managers to make decisions, and to exchange and cooperate with others. When a task cannot be completed, the system should also be able to seek partners to undertake the task, and to provide information services for other partners.

4) *Agility*. This is the ability that allows every entity in a system to make decisions independently and complete the task quickly. A manufacturing system consists of a number of manufacturing resources and entities, and it includes a complex material flow and an information flow. Its logical relationship is very complex. The information flow in a production planning and control system is required to send requests from bottom to top, and to send commands from top to bottom. By contrast, an interactive negotiation can be conducted in the same layer; an independent local decision can be made automatically; all entities jointly serve the common global goal, and accomplish the production planning and control task in coordination.
5) *Flexibility*. This is the ability of a system to respond rapidly to the dynamic environment. The characteristics of a modern manufacturing system are a distributed structure and dynamic events. All the related function modules are combined by integrating various heterogeneous information modules, physical layer devices, material information, and indicators in the control layer to generate a highly flexible whole module in order to manage a manufacturing system quickly and easily, and to ensure that it is easy to develop, debug and maintain a manufacturing system.

3.4 Multi-Agent-Based Push-Pull Production Planning and Control System (MAP4CS)

A MAP4CS, which consists of a push production planning subsystem, a push-pull production scheduling subsystem, and a pull production control subsystem is developed in this section. The production planning, scheduling and control system generated is not only required to make local decisions independently in the management area, but should also work together to support the global objective in order to complete the overall management tasks. The features of implementing the system include various data formats, multiple decision variables, frequent information communication, and complex information transmission paths. Agent technology is a method with adaptability, intelligence and self-organization. In a continuously variable and unpredictable environment, an Agent-based production planning and control system can rapidly adapt to sudden changes by handling various data rapidly and efficiently. Therefore, a MAP4CS is presented in this book to improve the flexibility and optimal performance of a manufacturing system.[36]

3.4.1 Mapping Methods

Agents are introduced in each function point to formulate an Agent mapping process in order to construct a MAP4CS. In this process, one kind of functional entity (including physical entity and logic function) is mapped (or encapsulated) to an Agent, which determines the granularity of an Agent, and affects the overall performance of a production planning and control system. The smaller the physical function scale is, the more Agents are in the system. The flexibility and adaptability of a system improve. However, the complexity of organization and control of a system increase, which may reduce the operating efficiency of a system. Currently there are no simple and efficient theory or mathematical methods to solve Agent mapping problems. A decomposition method is adopted in practice, which depends on the intuition and experience of a system planner. Since the system planning process is an innovative activity, this dependency is inevitable. The mapping methods can be classified as follows:

1) *Function mapping method*. In this method, an Agent is used to encapsulate system function modules, for example, order acquisition, planning, scheduling, materials handling, transportation management, product distribution and so on. There is no clear correspondence between an Agent and a physical entity. Agents generated by decomposing functions

must share some state variables with others, which may lead to consistency problems and unnecessary interactions.

2) *Physical mapping method*. In this method, an Agent is used to represent a physical entity in the real world, for example, worker, machine, product, component and process. The correspondence between an Agent and a physical entity is very clear. Agents generated by physical decomposition independently define and effectively manage a set of state variables respectively, which reduces interactive information.

Although it is hard to implement the function mapping method, it has shown good performances for large-scale complex systems. Therefore, the physical mapping method is adopted, while the function mapping method is supplemented to provide system level service. The entire process is conducted in the recursive format. Firstly, the physical mapping method is used to obtain its corresponding Agent entity. The whole system is regarded as a large Multi-agent System, and then it is decomposed into some subordinate Multi-Agent Systems, and subsystems can be further divided into smaller Agents. If an Agent granularity is satisfied, then terminate the process. Finally, each Agent plays an important role in the system. In the decomposition process, some Agents only temporarily exist, and may not be the final Agents in the system. The final Agents needed by the system are those that play important roles in the system.

3.4.2 Functions of a Hybrid Push-Pull Production Planning and Control System

Functions of a production planning and control system are collected as the prerequisite to conduct mapping to a MAS. By analyzing the transaction flow of a hybrid push-pull production planning and control system, functions of a production planning, scheduling and control system are further planned to generate specific functional requirements as shown in Figure 3-7.

3.4.2.1 Functions of a Push Production Planning Subsystem

A push production planning subsystem is responsible for managing orders and uncertainties, and planning manufacturing resources according to orders' requirements. Functions of this subsystem are presented as follows.

Time scale	Content	Function	System name
Long time scale (Ten-day/week)	Object: orders, predicted demand Resource: critical resource Objective: decide the amount of products to be produced in a period	What to do	Push production planning subsystem
Medium time scale (week/day)	Object: product tasks Resource: all the machine Objective: allocate resource to product tasks	How to do	Hybrid push-pull production scheduling subsystem
Short time scale (day/shift)	Object: process tasks Resource: all the machines Objective: collect data and manage the production process	Feed back completion status	Pull production control subsystem

Figure 3-7 Management layers and contents of a production planning and control system.

1) *Order management*. Customer orders are collected; orders' requirements are transformed to product tasks; completion status is feedback.
2) *Production plan management*. Workloads of key resources in the planning horizon are analyzed; types and quantities of final products in customers' demand are decided; the delivery due dates of the various products are determined.
3) *Planning layer collaborative management*. Resources status information and changed information in the planning horizon are collected and converted into resource workload information. All of these are completed by collaboration amongst Agents to provide information to optimize the production planning process.

3.4.2.2 Functions of a Hybrid Push-Pull Production Scheduling Subsystem

A hybrid push-pull production scheduling subsystem is responsible for allocating products to resources. Its major features include:

1) *Production scheduling*. Product tasks are allocated to machines; specific resources and completion times are determined to formulate the scheduled production plan while considering the constraints related to the processes of various tasks and the maximum capacities of manufacturing resources

2) *Rescheduling*. There are various dynamic events in a manufacturing system. The related real-time information is collected to implement the rescheduling process to update the scheduled production plan in order to effectively deal with these dynamic events.
3) *Scheduling layer collaborative management*. In view of the collaborative requirements of internal production resources, the real-time status information of each resource is collected by collaboratively managing Agents to support both the production scheduling process and the production rescheduling process.

3.4.2.3 Functions of a Pull Production Control Subsystem

A pull production control subsystem is responsible for releasing the production plan, collecting data and visualizing the real-time production data to manage the entire production execution process. Functions of this subsystem are presented as follows.

1) *Release the production plan*. A dispatch list for machines and workers in a given period is developed according to the production plan obtained by the production scheduling process.
2) *Data acquisition*. The real-time status data of machines, manufacturing times and product quality are collected to support production tracking and monitoring activities.
3) *Production monitoring*. The data collected is visualized; the processing tasks are monitored; responding to the results of production execution and to sudden changes; the fault information is sent to the planning layer and the scheduling layer; assisting in adjusting production task operation arrangement. This serves to monitor the production process, rectify deviations and support making decisions.
4) *Tracking production process*. The information collected in the data acquisition process is collated and classified. It includes the production progress tracking activity, the material tracking activity, and the quality tracking activity to meet the tracking requirements of the production process.
5) *Material management*. The movements, buffers and storage of materials are managed to provide a data base for the material

data retrospect. It can display and provide material data according to an enterprise's specific requirements, which accurately reflect the material handling situation of the production process. This function also involves constructing real-time data interfaces for all materials, and supporting production scheduling and rescheduling activities.

6) *Production performance analysis.* The latest performance evaluation report for an actual manufacturing process is formulated. The performance evaluation indicators are determined to reflect the production execution status while considering the production process evaluation requirements of each plant in the production process.
7) *Resource management.* Equipment and tools maintenance activities are tracked and guided; the accurate data of maintenance activities are recorded and analyzed statistically to support decision-making and to decide the number of maintenance operators and the processing time.
8) *Quality management.* The quality information generated in the production process is recorded and analyzed to control the product quality.

Moreover, some functions (e.g., basic production information management) are shared among subsystems in a hybrid push-pull production planning and control system. This includes basic information about components, product structures, processes and workstations. This information is presented as follows:

1) *Material code management.* Each material is given a unique code to manage material resources.
2) *BOM (bill of material) management.* The technical file of a product structure is defined to guide the planners to formulate production plans and production schedules.
3) *Process management.* The processing steps and the operation sequence of manufacturing and assembling of products are presented. These contain the operation sequence of a product, the processing machine for processing each process, and the operation processing time for processing each process.
4) *Resource fundamental data management.* Each resource is given a unique code to manage the fundamental data of machines and tools, which is a basic unit of production progress arrangement, capability calculation, and computation cost.

5) *Calendar management*. It is used to manage the factory calendar; it includes general holiday mode, working hours, overtime setting and so on.

3.4.3 Structures of a MAP4CS

The physical mapping method and the function mapping method are integrated to complete the Agent mapping process by using the recursive format in order to develop a MAP4CS. The basic procedure is presented below:

1) *Complete function mapping according to the hierarchical structure of a hybrid push-pull production planning and control system*. Then a Multi-Agent hierarchical production planning, scheduling and control system can be obtained.
2) *Complete physical mapping for resources in the production planning, scheduling and control process*. Then Agents including critical resource capacity management Agents, resource capacity management Agents, equipment management Agents and material management Agents in the enterprise layer, the plant layer, and the workshop layer can be obtained.
3) *Decompose tasks in a production planning, scheduling and control system*. Then order/product demand management Agents and task management Agents can be obtained.
4) *Mapping the specific functions of a production planning, scheduling and control system to Agents*. Then the collaborative planning Agent of a push production planning subsystem, the collaborative scheduling Agent of a hybrid push-pull production scheduling subsystem, the collaborative dispatching Agent of a pull production planning subsystem, the data acquisition Agent, the production process tracking Agent, the production monitoring Agent, the quality management Agent, and the alarm Agent can be obtained.
5) *Determine other auxiliary function Agents according to system requirements*. For all the Agents mentioned above, a set of Agents are grouped by their characteristics to form a Multi-Agent System with a clear organization structure. A Multi-Agent fundamental information management system, which consists of a process management Agent, a resource fundamental data management Agent, and a BOM management Agent, is developed by considering the fundamental

information (e.g., basic resource information, material code information, process information and BOM information) of a manufacturing system required in all the layers.

By mapping a production planning and control system to a Multi-Agent System, the structure of a MAP4CS is established, as shown in Figure 3-8.

The structures of a Multi-Agent System can be classified into three classes: hierarchical structure, federal structure, and complete autonomous structure. A hybrid hierarchical-autonomous structure is adopted by a MAP4CS to construct a Multi-Agent-based hybrid push-pull production planning and control system. The structure of a MAP4CS is composed of a planning layer, a scheduling layer, and a control layer. Each layer has its

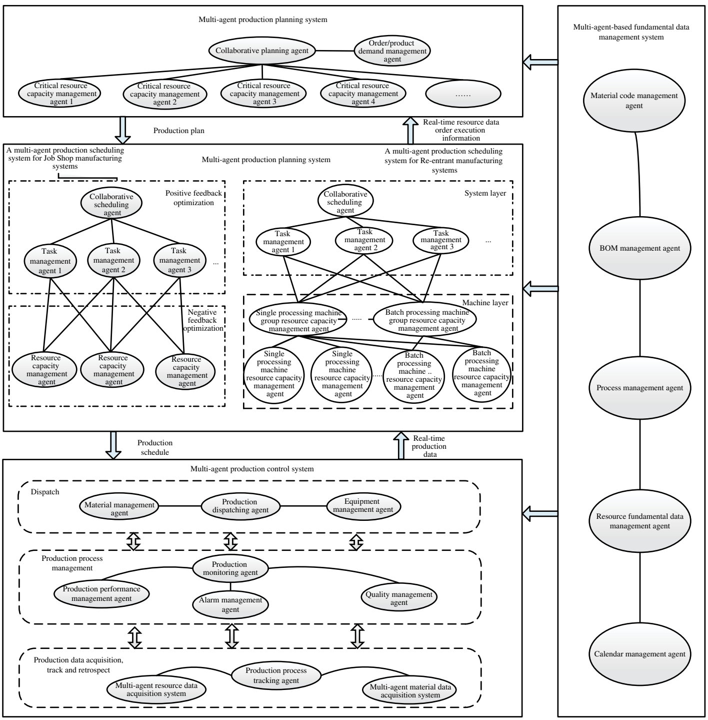

Figure 3-8 The structure of a MAP4CS.

independent internal structure to its corresponding intelligent management entity. Different layers work together to complete a series of activities; different internal collaborative decision strategies are used to optimize each layer. A complete autonomous structure is employed by the fundamental information management Agent and other Multi-Agent Systems. A MAP4CS has the following characteristics.

1) *Modularity*. This means that each function module has its independent features. In this structure, each Agent works and functions independently, such as resource management, production planning, production scheduling, and production control are relatively separated and greatly unified in one system. The management process is modularized so as to reconstruct the workshop management system according to environment and resources. Agents are introduced in different decision nodes to combine artificial intelligence techniques with manufacturing systems. Since different events are processed independently, each function of the hybrid push-pull production planning and control system should be realized by an Agent or a well-organized Multi-Agent System.
2) *Intelligence*. This means that each Agent has its own knowledge base and the ability to solve problems based on the knowledge base. In the system, each Agent solves local problems based on knowledge and collaborates with others to complete the overall task.
3) *Dynamic*. This refers to the dynamic nature of the organization structure of a Multi-Agent System. A hybrid push-pull production planning and control system can dynamically organize manufacturing resources to complete the production tasks to mostly satisfy requirements by using the existing manufacturing resources according to market opportunities and customer orders.
4) *Flexibility*. This refers to the sensitivity of a Multi-Agent System to external changes. The performance of a hybrid push-pull production planning and control system depends greatly on the speed of the feedback system. There is no strict control relationship among different layers of a hierarchical structure, but a gradually thinning decision relationship exists. Each layer has a relatively independent ability to make

decisions. In general, when a down-level adjusts due to changes, it doesn't need to feed back to the upper level; it collaborates with redundant network resources to make the system more sensitive to changes, so as to improve the flexibility of a hybrid push-pull production planning and control system.

5) *Autonomy*. This means that an Agent has the independent ability to deal with processing flow and to control logic activity. In a MAP4CS, orders are assigned at the production planning stage, product tasks are scheduled in the production scheduling stage, and a Multi-Agent control layer is responsible for tracking and collecting data in the production execution process. In this process, each Agent with its corresponding processing process, indicators and logical control ability collaborate with each other to improve the autonomy of a hybrid push-pull production planning and control system.
6) *Open*. This refers to the scalability of a system. Rather than using an organization-unit-oriented hierarchical structure, a decision-making process–oriented flat hierarchical structure is adopted by a Multi-Agent System. Indeed, the flat structure reduces decision-making areas; it improves the scalability of the system by integrating with a complete autonomous structure, which satisfies requirements for the openness of a hybrid push-pull production planning and control system.

3.4.4 The Running Model of a MAP4CS

The running process of a MAP4CS is completed by using a Multi-Agent production planning system, a Multi-Agent Job Shop production scheduling system, a Multi-Agent re-entrant production scheduling system, a Multi-Agent production control system, a Multi-Agent material data acquisition system, a Multi-Agent resource data acquisition system, and a Multi-Agent fundamental data management system. The running model is illustrated in Figure 3-9. When customer orders are accepted by a manufacturing system, production plans, production schedules, dispatch lists and production control instructions are generated by collaborating a Multi-Agent production planning system, a Multi-Agent production scheduling system,

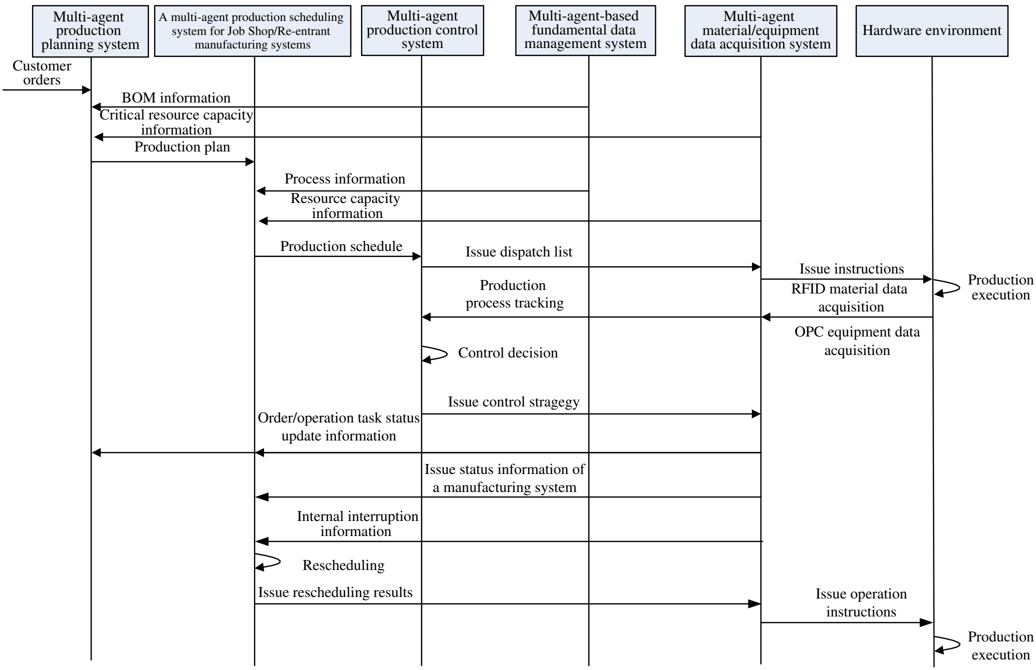

Figure 3-9 The running model of a MAP4CS.

a Multi-Agent production control system, a Multi-Agent fundamental data management system, and a Multi-Agent data acquisition system to guide the production execution process. The equipment status information and the material production process information are collected by using a Multi-Agent material data acquisition system and a Multi-Agent resource data acquisition system to support the production planning and control activity. When dynamic events (e.g., rush orders, machine breakdown) happen in a manufacturing system, a Multi-Agent production scheduling system will make a rescheduling decision. And a Multi-Agent production control system will guarantee the system stability by re-planning production control instructions.

3.4.5 Behavior Models of a MAP4CS

A MAP4CS can respond rapidly to market demand through mutual collaboration, achieve effective data feedback, and support the transparency of a production planning and control process. Behavior models of a MAP4CS are presented below.

3.4.5.1 Multi-Agent Push Production Planning System

As shown in Figure 3-10, a Multi-Agent push production planning system is used to determine the amount of products to be produced in a given period according to customers' orders and predicted demands, and to output the order delivery time, which is the first step to determine the delivery due date in a production

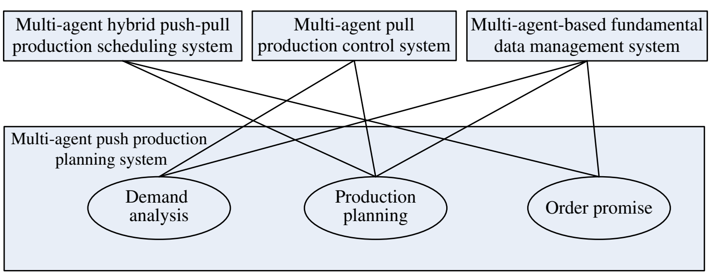

Figure 3-10 Use case diagram of a Multi-Agent push production planning system.

planning and control system and also to solve the upper management problem in a planning and control process. The production planning activity aims to determine a reasonable product task allocation scheme to make the production plan as stable as possible. Therefore, a Multi-Agent push production planning system should consider sudden changes in customers' demand; a production planning method is developed according to requirements to balance the production process. A balanced production plan is generated by considering changes of demand to guide the production process.

3.4.5.2 Multi-Agent Hybrid Push-Pull Production Scheduling System

As shown in Figure 3-11, a Multi-Agent hybrid push-pull production scheduling system is used to arrange product tasks, allocate machines, and determine completion times to guide the actual production process according to a specific production plan. In terms of dynamic events in the production process, a hybrid push-pull strategy is adopted in a Multi-Agent-based production scheduling system. In the periodic rolling phase, a push production scheduling model is developed, based on results of the production planning activity to improve the stability of a manufacturing system. In the dynamic environment, a pull rescheduling strategy is used to adjust the original production schedule to improve the adaptability of a manufacturing system to deal with dynamic events in the manufacturing system.

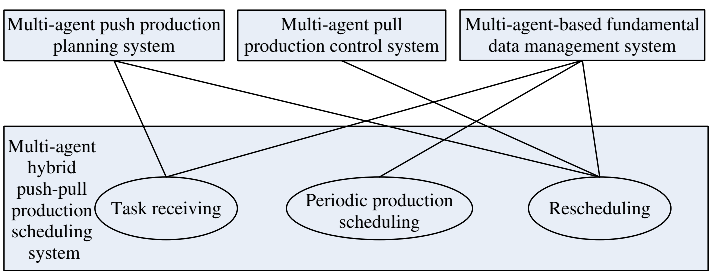

Figure 3-11 Use case diagram of a Multi-Agent hybrid push-pull production scheduling system.

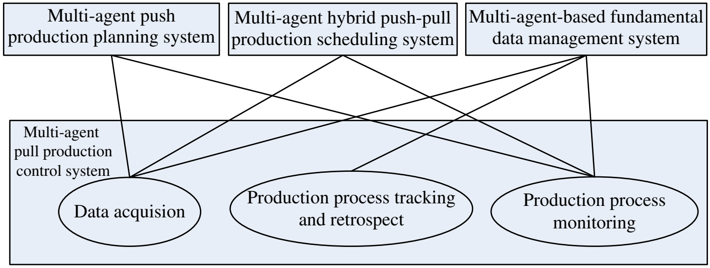

Figure 3-12 Use Case Diagram of a Multi-Agent Pull Production Control System.

3.4.5.3 Multi-Agent Pull Production Control System

As shown in Figure 3-12, a Multi-Agent pull production control system is used to visualize the production process by collecting material/equipment real-time data, to provide real-time data to support the Multi-Agent planning and scheduling system. Therefore, the running process of a Multi-Agent pull production control system is to collect real-time data in order to track and monitor the production process.

3.4.5.4 Multi-Agent Fundamental Data Management System

As shown in Figure 3-13, a Multi-Agent fundamental data management system is used to manage material code, BOM, equipment fundamental data and process information. Therefore, a Multi-Agent fundamental data management system is adopted to provide basic data to support other Multi-Agent systems in the production planning and control process.

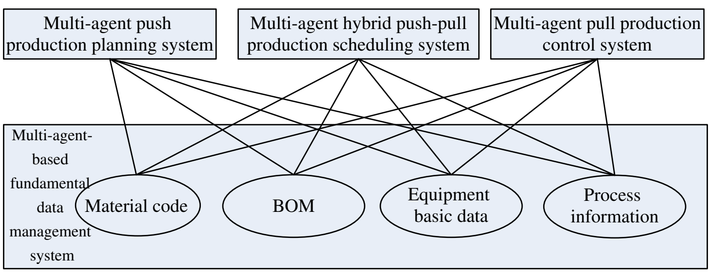

Figure 3-13 Use case diagram of a Multi-Agent fundamental data management system.

3.4.6 The Interactive Model of a MAP4CS

In order to run a MAP4CS effectively and stably, the interactive protocols and the communicative protocols between Agents should be defined. Since the interactive protocol between Agents directly reflects the purpose and rule of interaction, and theaims of Agents are different in different layers, different interaction protocols are adopted in different systems. In this book, Agent interaction protocols for specific problems are presented in Chapter 4 to Chapter 9 in order to optimize and solve problems.

The basis of communication between Agents in a Multi-Agent System is the communication protocol, which is independent from any one Agent. Each Agent deals with messages through a common message interface, which is independent from the internal data structure of an Agent. For the MAP4CS presented in this book, the knowledge query and manipulation language are adopted by all the messages. Only for some messages with remote service request, they are required to be transformed into their corresponding remote object requests, and to be issued by a remote service interface. This is presented in detail in Chapter 2.

Since the blackboard system is an expansion of agendas of traditional artificial intelligent systems and expert systems, it supports information transmission through appropriate communication protocols.[37] In a MAP4CS, a blackboard provides a common workspace as an information carrier, on which Agents can exchange information, data and knowledge. Agents can also write information on the blackboard; other Agents can read it.

In a blackboard system, each Agent has a number of blackboards. When one Agent exchanges information with another Agent, one can write data on the blackboard, and another reads it, and so communication does not occur directly between Agents. In the implementation process of a blackboard system, a blackboard can be defined as a class, and can derive many different classes by inheritance. An Agent finds the right class of a blackboard object through its properties. Typically, each Agent has a blackboard object. Then Agents will exchange information through the blackboard. A Multi-Agent System can also contact other systems through a blackboard.

Agents can access the blackboard at any time, and read the information if it is updated. The information would expand more and more through accumulation; if it is not effectively maintained and managed, it will not only occupy resources, but also become more difficult to search. Hence, it is necessary for a blackboard system to delete the old information when its capacity exceeds a certain amount, and to set a valid time limit for the published information.

The main parameters concerned in a production planning and control system consist of resource capacity information, the unit product processing time information, task information and so on. This information is presented as follows:

3.4.6.1 Resource Capacity

The expression (p, c) is used to express the resource capacity information on the blackboard, where p is a product identifier (i.e., the type of a product, rather than the product instance), and c is a resource capacity. Each type of resource displays its capacity information on the resource panel. The resource capacity information can also be unified on a blackboard as shown in Figure 3-14.

The resource capacity information should be updated in real-time so as to ensure the accuracy of the information in the collaboration process. In the resource capacity layer, the content of issued information is shown in Figure 3-15.

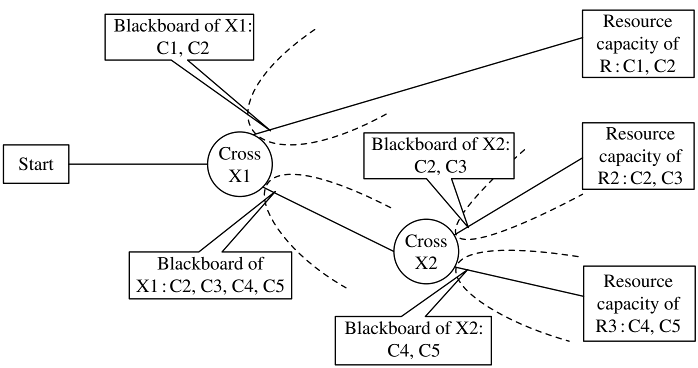

Figure 3-14 Resource capacity information blackboard of an Agent.

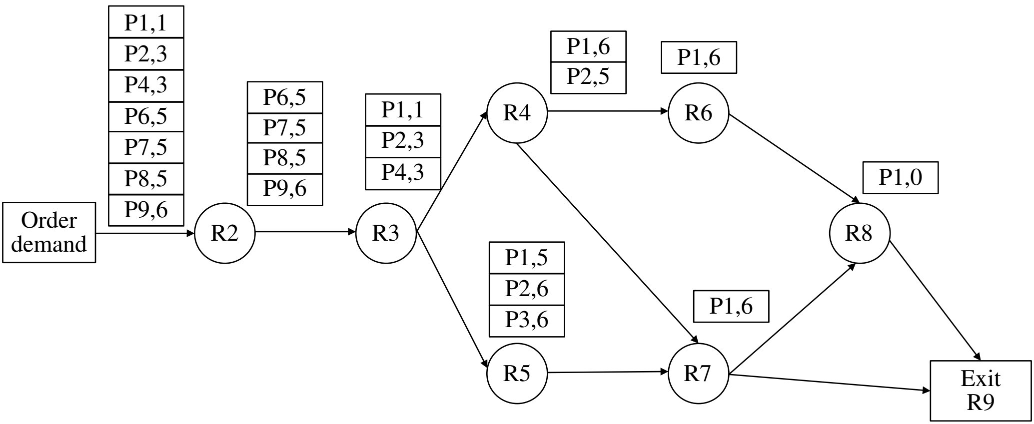

Figure 3-15 Resource capacity information.

3.4.6.2 Unit Product Processing Time Information

By measuring the unit product processing time in different resources, the unit product processing time information is then issued throughout the whole resource network. For each resource, a blackboard for its corresponding Agent is established, in which the information is expressed by the format (destinationResourceID, currentResourceID, transportTime). Where destinationResourceID represents the number of an objective resource, currentResourceID represents the number of the current resource, transportTime represents the unit product processing time. Figure 3-16 illustrates the unit product processing time information blackboard. The unit product processing time required in each resource is transmitted in this way so as to ensure that each resource has a corresponding processing time information identifier.

3.4.6.3 Task List

The task list in a blackboard is expressed by the format (orderID, productID, productAmount), where orderID is order No., productID is material No., productAmount is the quantity of external demand.

Each Agent places various blackboards in their corresponding positions. A resource management Agent is taken as an example, the blackboards are placed on each type of resources. A resource management Agent can write real-time resource information on a blackboard so as to display the latest information in its list. When a resource allocation Agent arrives and uses resources for an advance, or changes the reserved resources due to repair and maintenance, the resource list should be updated. When a resource in a resource management Agent is ordered, the Agent will perform the update operation, as shown in Figure 3-17. If the task list changes, the resource management Agent will update the information. The resource management Agent calculates the actual occupied time and the amount of resources, and then it updates and releases the information.

When a resource breaks down, a resource management Agent triggers an update, then the information in the resource information list is updated. The time information blackboard accurately reflects the situation. When a resource breaks down, its related capacity information will be updated, and the updated

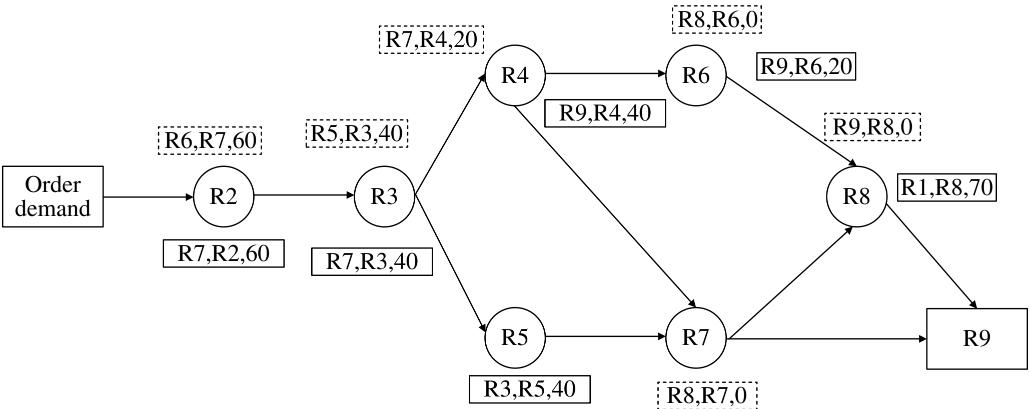

Figure 3-16 Unit product processing time information blackboard.

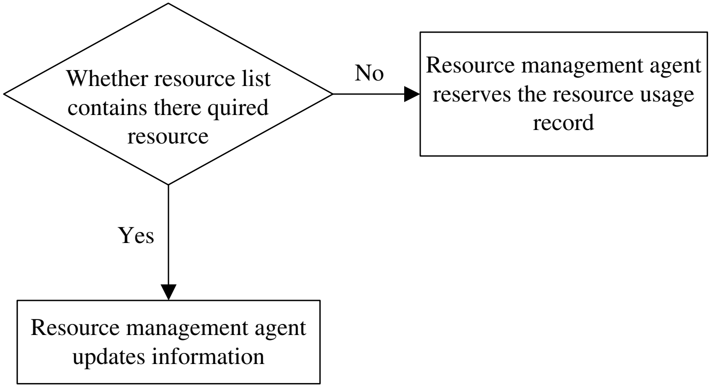

Figure 3-17 Update resource agent information list.

information is transferred to other relevant Agents. Therefore, the task list related to these resources should be updated, and those tasks whose starting time is later than the breakdown time should be removed. Once the resource is recovered, it will trigger another update, and notify other Agents to respond to the new task information.

3.5 Conclusion

Firstly, the basic concept of manufacturing systems has been presented in this chapter, and manufacturing systems have been classified from three aspects, that is, the manufacturing process, organization and the material handling mode. Secondly, production planning and control systems have been analyzed, and the basic activities in the systems have been presented. Production planning and control systems can be classified as those based on push mode and those based on pull mode. By analyzing various methods and current systems, the advantages and disadvantages of existing systems have been summarized. Then in view of the practical requirements of current manufacturing systems, a hybrid push-pull production planning and control system has been proposed. Finally, a Multi-Agent–based hybrid push-pull production planning and control system has been developed. The structure characteristics, behavior models, running models and information interaction modes of the system have been further illustrated by analyzing various Agents and their functions.

References

[1] Li Peigen. *Manufacturing System Performance Analysis Modeling—Theory and Methodology*. Wuhai: Huazhong University of Science and Technology Press, 1998.

[2] Guo Jun, Cao Yan, and Liu Ning. Manufacturing System Development Strategic Implementation Mode Selection. *Yizhong Technology*, 2006, 4: 116–118.

[3] Liu Fei, Yang Dan, and Chen Jin. *Manufacturing System Engineering*. Beijing: National Defense Industry Press, 1995.

[4] *Cihai*. Shanghai: Shanghai Lexicographical Publishing House, 2009.

[5] Hsue-shen Tsien. *Discussion on System Engineering—Hsue-shen Tsien's System Science Thought Library*. Shanghai: Shanghai Jiao Tong University Press.

[6] Agnes, M. *Webster's Dictionary*. Shenyang: Liaoning Education Press, 2004.

[7] Japanese Industry Association. Industrial Standardized Method. 1949.

[8] Chisholm, A. W. J. Nomenclature and Definitions for Manufacturing System. *Annals of the CIRP*, 1990, 39(2): 735–744.

[9] Chryssolouris, G. *Manufacturing System: Theory and Practice*, New York: Springer-Verlag, 1992.

[10] Winkler, J., and Mey, M. Holonic Manufacturing System. *European Production Engineering*, 1994, 19(3): 10–12.

[11] Tharumarajah, A., Wells, A. J., and Nemes, I. Comparison of the Bionic, Fractal and Holonic Manufacturing System Concepts. *International Journal of Computer Integrated Manufacturing*, 1996, 9(3): 217–226.

[12] Mill, F., and Sherock, A. Biological Analogies in Manufacturing. *Computers in Industry*, 2000, 43(2): 153–160.

[13] Ueda, K., Vaario, J., and Ohkura, K. Modeling of Biological Manufacturing Systems for Dynamic Reconfiguration. *Annals of the CIRP*, 1994, 43(1): 343–346.

[14] Iwata, K., and Onosato, M. Random Manufacturing Systems: A New Concept of Manufacturing Systems for Production to Order. *Annals of the CIRP*, 1994, 43(1): 379–383.

[15] Lee, G. H. Re-Configurability Consideration Design of Components and Manufacturing Systems. *International Journal of Advanced Manufacturing Technology*, 1997, 13(5): 376–386.

[16] Kumar, P. R. Re-Entrant lines. *Queuing Systems*, 1993, 13(1–3): 87–110.

[17] Li Huaizu. Production Planning and Control. Beijing: China Science and Technology Press, 2001.

[18] Zhou Shuiyin, and Chen Rongqiu. Internet Based Virtual Enterprise Resource Planning V-ERP. *Computation Engineering and Application*, 2001, 37(6): 20–22.

[19] Liu Fei, Yang Dan, Chen Jin. *Manufacturing System Engineering*. Beijing: National Defense Industry Press, 1996.

[20] Bongaerts, L., Monostori, L., McFarlane, D., et al. Hierarchy in Distributed Shop Floor Control. *Computers in Industry*, 2000, 43(2): 123–137.

[21] Zhang Jie. Research on Workshop Control Model. Postdoctoral research report of Huazhong University of Science and Technology. Wuhai: Huazhong University of Science and Technology, 1999.

[22] Zhang Jie, Wang Yu, Liu Shiping, and Li Peigen. Multi-Agent Based Workshop Control System Framework. *Journal of Huazhong University of Science and Technology*, 2001, 29(11): 9–12.

[23] Liu, S. P., Zhang, J., and Li, P. G. Study on Multi-Agent Based Shop Floor Manufacturing and Control System. *Proceedings of International Conference on Advanced Manufacturing Systems and Manufacturing Automation*, Guangzhou, China, 2000, 883–886.

[24] Chan, F. T. S., and Zhang, J. A Multi-Agent-Based Agile Shop Floor Control System. *International Journal of Advanced Manufacturing Technology*, 2002, (19): 764–774.

[25] Finin, Labrou,Y. and May, J. KQML as an Agent Communication Language, In Jeradshaw, editor, *Software Agents*. Cambridge, MA: MIT Press, 1997.

[26] Hopp, W. J., Spearman, M. L. Commissioned Paper to Pull or Not to Pull: What Is the Question? *Manufacturing & Service Operations Management*, 2004, 6(2): 133–148.
[27] Chen Rongqiu. *Production Planning and Control—Concept, Method and System*. Wuhai: Huazhong University of Science and Technology Press, 1995.
[28] Price, W., Gravel, M., and Nsakanada, A. L. A Review of Optimization Models of Kanban-Based Production System. *European Journal of Operational Research*, 1994, 75(1): 1–12.
[29] Wang Shijin. Research on Autonomy and Negotiation Mechanism Based Flexible Manufacturing System Intelligent Scheduling Technology. Shanghai Jiao Tong University PhD dissertation, 2008.
[30] Cott, B. J., and Macchietto, S. Minimizing the Effects of Batch Process Variability Using Online Scheduling Modification. *Computers and Chemical Engineering*, 1989, 13(1): 105–111.
[31] Kanakamedala, K. B., Reklaitis, G. V., and Venkatasubramanian, V. Reactive Schedule Modification in Multi-Purpose Batch Chemical Plants. *Industrial Engineering and Chemical Research*, 1994, 33(1): 77–90.
[32] Nelson, R. T., Holloway, C. A., and Wong, R. M. Centralized Scheduling and Priority Implementation Heuristics for a Dynamic Job Shop Model with Due Dates and Variable Processing Time. *AIIE Transactions*, 1977, 19(1): 95–102.
[33] Fang Jian, Xi Yugeng. Genetic Algorithm Based Rolling Scheduling Strategy. *Control Theory and Application*, 1997, 14(4): 1–8.
[34] Zhu, Q., and Zhang, J. Agent Based Production Planning and Scheduling System for Networked Manufacturing System. *The 8th International Conference on Supply Chain Management and Information Systems*, 6th–8th October 2010, Hong Kong, China, 132–136.
[35] Wang W.H., Zhu Q., and Zhang J. An Available-to-Promise System for Network-Manufacturing. *Advanced Materials Research*, 2011, 268–270: 303–308.
[36] Wang Shiming. Anti-Interference of Microcomputer Control System. *Machine Tool and Hydraulics*, 1998, 4: 1–2.
[37] Zhao Huoming. Research on Multi-Agent Based Network Manufacturing System. Hefei University of Technology, PhD Dissertation, 2000.

4

Multi-Agent-Based Production Planning for Distributed Manufacturing Systems

4.1 Introduction

With the rapid development of IT and network technologies, the market tends to be global, so that manufacturing resources are available all over the world. Meanwhile, customers' demands tend to be personalized and diversified. As a response to these demands, customization, large product variation, small batches and quick delivery become the rule-of-thumb criteria for the evolution of new manufacturing systems. Therefore, a new system called distributed manufacturing has been proposed, in which various companies cooperate collaboratively. To deal with unpredictable events in the global market environment, the production planning process should optimize long-term task allocation while considering the impacts of uncertainties (i.e., the push production planning process presented in Chapter 3). In terms of the characteristics of a distributed artificial intelligence–based Agent technology such as personification, autonomy, adaptability, intelligence and collaboration, it can be used to formulate production plans for distributed manufacturing systems, as well as to improve the flexibility and agility of the medium to plan over the long term for uncertainties.

4.2 Production Planning for Distributed Manufacturing Systems

4.2.1 Distributed Manufacturing Systems

With the changeover of enterprises' external environments, internal organization and management concepts, and the development of IT and enabling technologies, the traditional "all-control", rigid, pyramid production organization mode is no longer applicable. Meanwhile organization modes with high flexibility, flat organizational structure and strong adaptability to changing market conditions are gradually attracting production managers' attention. More enterprises are transforming to the customer demand–oriented, core competencies–supported, and collaboration-based distributed manufacturing mode. In this process, some large enterprises are gradually being divided into a number of small subcompanies, each of which has its own core business and focuses on producing specific types of products. On the other hand, some SMEs (small and medium-sized enterprises) tend to share their information, technology and knowledge to generate a network-like or chain-like distributed structures in the network environment in order to implement distributed collaborative manufacturing to meet specific customers' demands. Figure 4-1 shows the typical structure of a network-like distributed manufacturing system, in which individuals assign decision-making abilities by coordinated capacities, creating strong flexibility and adaptability. Currently, with increased varieties for consumer demands, accelerated technology updates, shortened product life cycles, and intensified enterprise competitions, distributed manufacturing systems have been developed rapidly. The distributed manufacturing system is now playing an increasingly important role in the economic globalization process;[1] it acts as a critical tool to improve the resilience, adaptability, flexibility, and robustness of manufacturing enterprises.

In contrast to traditional single factory manufacturing systems, distributed manufacturing systems have the following characteristics:

1) *Distributed autonomous decision-making*. This model has been adopted by a number of manufacturing systems

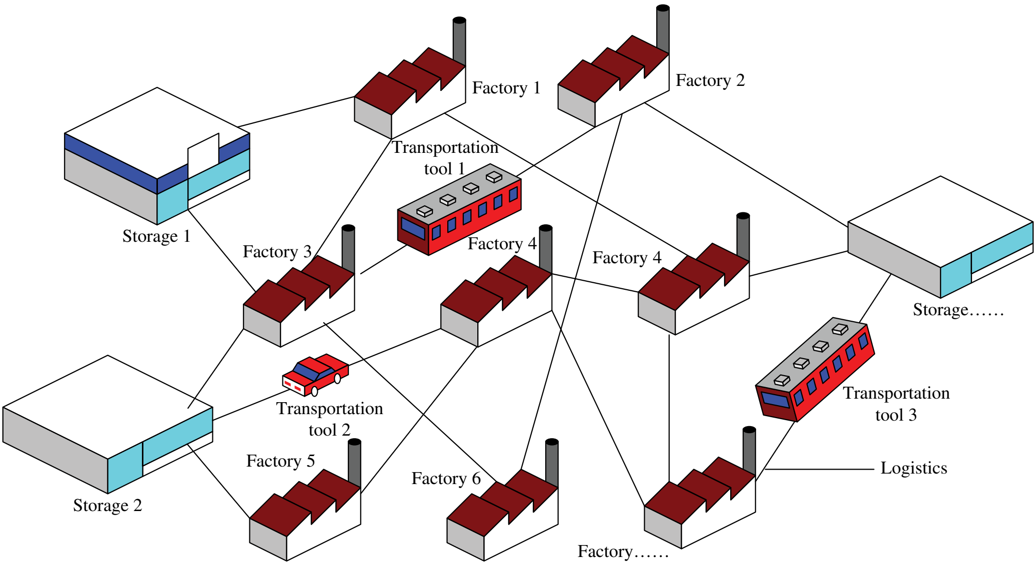

Figure 4-1 A distributed manufacturing system.

proposed in recent years, and the distributed structure is used by these manufacturing systems. The basic cell of a manufacturing system contains a set of manufacturing resources that are able to accomplish tasks independently. Each resource manager in a system has autonomy and can make decisions alone.

2) *Collaboration.* Various advanced manufacturing technologies have been developed by factories in distributed manufacturing systems to enhance self-competitiveness in order to deal with the ever-changing business environment and internal organization. However, the performance of an enterprise depends not only on manufacturing technology and management models, but also on their collaborative manner. Enterprises improve their manufacturing flexibility and compress lead time through collaboration, which further improves their response ability and revenue.

Distributed manufacturing systems have been proposed for distributed autonomous decision-making and collaboration. With the development of manufacturing science and technology, as well as the integration with techniques of computer, communication, network, automation and artificial intelligence, the composition of modern manufacturing systems has achieved a qualitative leap. New organization structures, their corresponding internal management, and collaborative modes should be adopted in order to better meet market demand, improve the performance of manufacturing systems and implement the flexibility, agility, scalability and reorganization ability, reliability and fault tolerance of manufacturing systems. Various components of manufacturing systems have collaborated effectively by constructing the organizational structure and collaborative manner that is adaptable to the current market environment to respond rapidly to market changes, to output in a timely manner the products needed by the market, reduce production costs, and improve machine utilization so as to obtain a better investment return. In terms of manufacturing technology and current status of manufacturing systems, new production planning approaches for distributed manufacturing systems are investigated in this section.

4.2.2 Features of Distributed Manufacturing Systems

It is hard to implement production planning, scheduling and control processes in distributed manufacturing systems due to their complex organizational structures As regards driving data, production planning results reflect what companies intend to produce, as well as when and how much to produce. Specifically, a production plan must arrange the product types, output speed, output time, labor, equipment configuration, inventory and other issues of manufacturing systems in advance according to the market demands and production capacity of an enterprise. Customers' orders, customers' predicted demands, available resources and management objectives must be taken into consideration by a production plan, which determines the quantity of products produced and the delivery time in the planning horizon, and balances the production demands and available resources by the rough capacity plan (RCCP). The production planning objects in manufacturing systems are usually the final products, that is the products on sale. A production plan is eventually executed by production sectors, and is regarded as a standard to evaluate the customer service level.

The main differences between a distributed manufacturing system and a traditional centralized single plant manufacturing system is in resource location and decision-making processes. Hence, the production planning process in distributed manufacturing systems is different from that in traditional manufacturing systems. The traditional production planning process is illustrated in Figure 4-2. Firstly, a production plan arranges the type and quantity of products to be produced each period in the future according to predictions and customer orders. Secondly, the plan must balance critical capacities. It focuses on calculating the capacities of critical work centers, and evaluating whether the existing personnel, equipment, plant, capital and other resources can meet the plan demand in order to arrange timely resources and ensure the production plan execution. The balance of critical capabilities is used to estimate whether the production plan could be carried out and determine where there might be bottleneck machines. Then a production plan is confirmed, and the production planning process is completed.

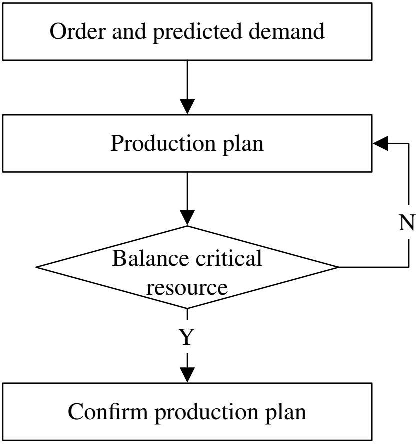

Figure 4-2 Traditional production planning process.

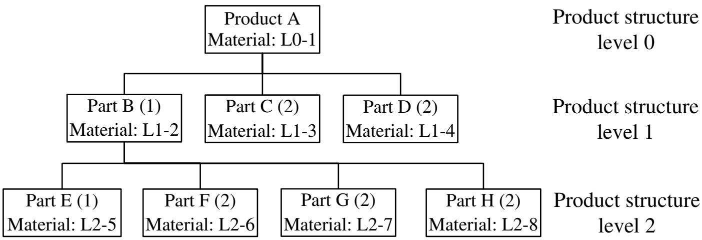

Figure 4-3 The structure tree of product A.

In a multi-plant distributed manufacturing system, since the issues related to critical capacities are more extensive, it is necessary to consider the capacity balance of distributed resources. Due to the complexity of the problem, capacities of plants to manufacture components must be confirmed and balanced. Next, we will give the definition of a production plan in distributed manufacturing systems and their differences from traditional ones by using an illustrative example. The bill of material (BOM) of product A is illustrated in Figure 4-3, where product A is the final product and contains five types of components, namely, B, C, D, E and F. In this product structure, the relationship between adjacent upper level and lower level is parent and children. For example, product A is the parent of components B, C and D, or components B, C and D are the children of product A. In the figure, data in brackets represent the number of children every parent needed, and every product, component or part in a node corresponds to a production plant. In this case, every plant should determine the supply capacity in each specific period in the production planning process. Therefore, the production planning process in distributed manufacturing systems is defined as follows: according to market demands and capacities of distributed enterprises, a plan determines the quantity of each final product that is produced in each specific period.

In contrast to the production planning process in traditional manufacturing systems, the planning process in distributed manufacturing systems changes a lot and adds the following features:

1) *High integration and synchronization*. Processing information sharing, quick response ability, and flexibility are the key factors to achieve synchronization. Those who cooperate can improve the performance of manufacturing systems and achieve quick response ability by visualizing manufacturing information. On the other hand, they might achieve tight integration and perfect information sharing through efficient organization of the manufacturing systems. As resources of current manufacturing systems become increasingly complex, the demand for flexibility and information visualization of manufacturing systems becomes higher. It is necessary to achieve an orderly organization at all levels, to enhance and optimize scheduling results.
2) *Collaboration*. Information in the manufacturing process is transformed in real-time through collaboration. A production plan should effectively regulate changes in demand and supply in each period so as to avoid production delays or excessive results from disorder production.
3) *Uncertainty*. It is known that plants in distributed manufacturing systems are autonomous and collaborative. However, there is competition among plants as they consider requirements of production efficiency and development. Therefore, the interaction and coordination between plants should be taken into consideration in the production planning process in order to achieve efficiency and meet customers' demands.

4) *Extensive information*. Nowadays it has been noted that the completion of an order involves the cooperation of several companies; the production plan is not limited to a single production entity; it is related to production activities with timing constraints among production members. In the traditional production planning process, information is composed of two parts: demand information and internal resource information. On the other hand, in a distributed manufacturing system, the resource information contains not only its own information but also the collaborators' information.

In summary, the production planning process in distributed manufacturing systems plays a very important role in the production management process. It develops a bridge between the inside (manufacturing) and the outside (marketing and sales) of an enterprise. Therefore, it is important to propose an effective production planning method to develop and manage a new manufacturing enterprise in the distributed manufacturing environment.

4.2.3 Production Planning Methods for Distributed Manufacturing Systems

The production planning process in distributed manufacturing systems is an important task for plants; the key to solve this problem is to design an optimization method. This optimization problem has attracted the attention of scientists in the field of applied mathematics, operations research and engineering technology; the field has seen many important achievements, while various optimization methods are still booming. In the past decades, production planning optimization methods began with operations research methods, including a variety of mathematical programming methods; then a variety of heuristic rules appeared while considering computation times; by the 1980s, artificial intelligence methods began to be used to solve production planning problems, including expert system, Multi-Agent Systems, intelligent optimization algorithms and other methods. Production planning optimization methods for distributed manufacturing systems can be classified into three major classes: operations research methods, heuristic methods and artificial intelligence methods.[2]

4.2.3.1 Operations Research Methods

Operations research methods mainly focus on solving linear programming and mixed integer linear programming problems, specifically the simplex method, the Benders decomposition method,[3] the branch and bound method,[4] the Lagrangian relaxation method[5] and so on. These kinds of methods have generated a mature theory, and many commercialized software packages have been developed, such as Cplex, Lingo and so on.

Mathematical programming–based production planning methods mainly use linear programming, mixed integer linear programming and dynamic programming to analyze planning problems.[6] Since mathematical programming methods belong to the accurate modeling methods, any changes of problem parameters may cause a model to fail in a different production environment. Linear programming is primarily used to determine the investment, number of products, demand for raw materials, number of machines needed, energy demands and work hours required, in order to obtain the maximum economic benefit. Hu Zexin[7, 8] established the mathematical programming model for a production planning problem to maximize the economic benefits, which included the constraints related to machine quantity and working hours required.

Mixed integer linear programming models are linear programming models with integer decision variables, for example, the number of machines and workers in manufacturing systems. While considering the availability of resources, equipment or tools in different stages, some decision variables are 0 or 1, and then the model becomes an 0/1 mixed integer linear programming model.[9] Sahindis and Grossmnna[10] considered the availability of equipment, and set the decision variables corresponding to the assignment of equipment to task as 0/1, then constructed a 0/1 mixed integer linear programming model to minimize the production cost.

Dynamic programming models are developed on the basis of optimization theory. The main idea is to decompose an optimization problem into multiple interrelated stages, a decision has to be made in each stage so as to achieve the best activity results. Emlaghrbay[11] considered the situation where the selection of available resources was not arbitrary at different stages; the decision depended on the current state and affected the future development. When decisions were made in various stages, a decision network was constructed, and then a dynamic programming model based resource allocation model was built. Dynamic programming models are mainly used to establish models for the dynamic process optimization problem with several stages in different periods.

The common feature of operations research methods is to seek an optimal or near-optimal polynomial time algorithm for production planning problems. However, with the expansion of the problem scale and constraint quantity, the computation time grows exponentially. Therefore, this kind of method is primarily suited to deal with small-scale optimization problems.

4.2.3.2 Heuristic Methods

The heuristic method is to use task-irrelevant information to simplify the search process, and the solving process is to systematically construct or find solutions, during which search space checking, possible solvability paths evaluation and searched paths record are included to form the problem solution. Panwalkar[12] reviewed 113 heuristic methods, and divided them into three categories: simple rules, composite rules and heuristic rules. Kanakamedala[13] et al. proposed a heuristic method for production planning problems, in which the Beam search method with the "minimal impact" heuristic rule was used for the pruning of a two-story tree. Together with Kusiak,[14] he constructed an optimization model for production planning problems in network distributed enterprises. He set every production entity as manufacturer or assembler, and the proposed model was to assign tasks to the assembly center to minimize the average transportation cost and inventory cost. A heuristic method was used in the solving process of the model, and experiment results showed the feasibility of the method.

Heuristic search methods provide some other advantages, such as easy implementation and low computation complexity, so it is suitable for complex dynamic environments. It has been attracting researchers' attention for quite a long time, and many new rules are emerging. This kind of method uses the knowledge and experience of specific problems and can generate a better solution in a short time. However, only a few attempts have been seen to evaluate the effect of solutions. Besides, it is necessary to investigate measures to improve efficiency and solve large-scale problems.

4.2.3.3 Artificial Intelligence Methods

In the field of production management, the Artificial intelligence (AI) method is a MAS-based optimization method which is widely used and has made a great progress.

In the 1980s, the MAS technology was gradually used for production management research areas along with its continuous research. Then the study of MAS enjoyed a boom after the 1990s with a diversification of the technology roadmap. Bussmann[15] studied the Agent-based distributed constrained heuristic search (DCHS) problem. He distributed the entire problem to each Agent, and carried out asynchronous and parallel searches. The solution has been validated in production planning problems in distributed Job Shop–type manufacturing systems. Dong Hongbin[16] viewed all functions and entities of a manufacturing system as Agents in the proposed MAS framework, and presented a price mechanism–based market model to express the negotiation between Agents. Then he proposed a contract network protocol-based distributed scheduling strategy for flexible manufacturing systems. Yanxiang[17] presented a knowledge-based task assignment method for distributed computer systems with an application of Agent technology. This study can serve as an important reference for the design of Agents in production planning processes.

One of the key problems in MAS-based production planning methods is the collaboration and negotiation between Agents. The purpose of deciding upon a collaboration/negotiation mechanism is to determine the joint problem-solving mechanism of Agents.

In today's global economy, the market demands stricter requirements for manufacturers in regard to product quality, supply response speed, agility and flexibility. This trend has encouraged companies to improve their competitiveness, no longer by acting alone, but together with other companies in the global competitive market environment. The production plan for distributed manufacturing systems is to achieve the overall objective of the enterprises. In order to realize these goals, it determines the production targets for all the enterprises of distributed manufacturing systems, balances these targets among the production factors of the manufacturing system, and globally arranges various resources in time and space.[18] Production planning problems have been endowed with new features in this new environment. It can easily be seen that the classical optimization methods have become nearly mature; however, they still cannot satisfy requirements of practical problem and computation precision. Some basic principles of the classical theory cannot be used for solving practical problems; they should be further reconsidered and expanded. Single operations research methods, heuristic methods or artificial intelligence technology can achieve satisfactory performance only with difficulty. Therefore, the integration of a variety of artificial intelligence technology-based methods is one of main directions to be considered for future research in the area of production planning problems.

4.3 Multi-Agent-Based Production Planning in Distributed Manufacturing Systems

Because the alliance formation of enterprises is often scattered in different areas, the production planning process in the current manufacturing system is based on a large system in the form of distributed network rather than one based on the traditional internal processing chain. Demand management must be associated with resource capacity to realize the interaction and integration of heterogeneous information.[19] Since a production plan is long-term, it should consider a variety of unknown conditions over a certain period to allocate resources. From another perspective, the uncertainty in the environment makes the production planning process more complex. Agent technology has good intelligence and distributed computing capability for such a complex distributed optimization problem. A production planning model for distributed manufacturing systems is presented in the following section to construct MAS solutions for this problem, in which a MAS is described, and the structural model and the behavioral model of each Agent are given.

4.3.1 A Production Planning Model for Distributed Manufacturing Systems

At the production planning stage, the potential impact of uncertain factors in a period should be considered to make decisions in order to deal with uncertain factors. The production planning results must absorb or resist the effects of uncertain factors to some extent, so as to be able to adjust production plan during the execution process. The general approach to deal with uncertain factors includes probability theory, fuzzy set theory, and interval algebraic method. Sensitivity analysis method comes from uncertain factor (i.e., statistics, prediction, estimation, and assumption) modeling, and it can be used to analyze the influence of the uncertain factor changeover on the model output.[20] The sensitivity analysis method determines the effects of uncertain factors through a passive behavior. The sensitivity analysis method does not consider uncertain factors in advance; the shortcomings of this method are that the results depend on the optimal solutions or the search method of optimal solutions. The stochastic programming method acquires the features of regular model structure and block structure.[21] Hence, the decomposition method is commonly used in calculations, which may lead to the rapid expansion of the model and cause difficulties in computation. The Fuzzy Programming method adopts the possibility programming approach.[22] It has been applied successfully in a wide variety of problems, and it usually is used where the parameters of objective function and constraints are uncertain. The robust optimization method is based on a scenario analysis and design issues.[23] Its purpose is to find a balance between the optimal solution and feasible solutions. A production planning model for distributed manufacturing systems is developed by using a scenario analysis–based robust optimization method.

4.3.1.1 The Basic Principle of Robust Optimization

The robust optimization model is presented on the basis of a typical mixed integer programming model. Assuming that there are three product lines that can produce the same kind of products. The operation costs of each assembly line, the production cost of a product, and the largest production capacity are

Table 4-1 Resource Information.

Resource	Operation Cost	Cost per Product	Maximum Capacity
Assembly line 1	100	10	800
Assembly line 2	200	4	1400
Assembly line 3	300	2	1000

shown in Table 4-1. A reasonable production plan should be formulated to manufacture 4000 products to maximize the profit.

In the Model, the capacities of assembly line 1, 2 and 3 are respectively x_1, x_2 and x_3. Three 0/1 variables y_1, y_2, and y_3, are used to represent whether each assembly line is ready to start. The mixed integer programming model has the following form:

$$\min \quad z = 10x_1 + 4x_2 + 2x_3 + 100y_1 + 200y_2 + 300y_3 \tag{4-1}$$

$$\begin{aligned} s.t. \quad & x_1 + x_2 + x_3 \le 4000 \\ & x_1 - 800y_1 \le 0 \\ & x_2 - 1400y_2 \le 0 \\ & x_3 - 1000y_3 \le 0 \\ & x_1, x_2, x_3 \ge 0, y_1, y_2, y_3 = 0/1 \end{aligned}$$

The mix integer programming model above can be described in the form of basic linear programming model as follows:

$$\max \quad z = c^T x + d^T y \tag{4-2}$$

$$s.t. \quad Ax = b \tag{4-3}$$

$$Bx + Cy = e \tag{4-4}$$

$$x, y \ge 0 \tag{4-5}$$

Where z is the objective function, x and y are decision variables. Formula 4-3 is an independent constraint, and A is a parameter matrix. Formula 4-4 is the constraints generated by interaction of decision variables, B and C are parameter matrixes, e is the vector constraints. Formula 4-5 is the non-negative constraints.

In 1995, Mulvey et al.[24] proposed a robust optimization method to solve large-scale optimization problems with uncertain factors. They combined the objective programming model with the problem data description method, and established a robust optimization model on the basis of the mixed integer programming problem. The set of possible scenario values is $\Omega = \{1,2,\ldots,S\}$; and the set of parameters corresponding to each scenario $s \in \Omega$ is $\{d_s, B_s, C_s, e_s\}$; the probability is P_s where $\sum_{s\in\Omega} p_s = 1$. The robust optimization model for the above problem is presented as follows:

$$\min \quad \sigma(x, y_1, \ldots, y_S) + \omega\rho(\delta_1, \ldots, \delta_s) \tag{4-6}$$

$$s.t. \quad Ax = b \tag{4-7}$$

$$B_s x + C_s y_s + \delta_s = e_s \quad \textit{for all } s \in \Omega \tag{4-8}$$

$$x, y_s \geq 0 \quad \textit{for all } s \in \Omega \tag{4-9}$$

The first part of formula 4-6 is the robustness of solutions, which is to express the lowest cost or risk objective function. The second part represents the robustness of the model, which indicates the penalty cost when some constraints cannot be satisfied, for example, the production constraint is not satisfied. δ_s is the error vector, which represents the degree that the actual value deviates from the constraint value. ω is a weighted parameter to express the ratio of solution robustness to model robustness. The model consists of two different components. One is the structure component that is fixed and not affected by noise and input; another is the control component that is influenced by input. In addition, two sets of variables are also included in the model. x is the decision variable, the optimal value of which does not depend on the realization of all uncertainties; y_s is the control variable, the optimum value of which depends on the realization of all uncertainties. Since the infeasible situation is inevitable, the penalty function is adopted to coordinate it. The selection of penalty functions is closely related to the problem background. Yu and Li[25] gave a further definition for the robustness of solutions, just as formula 4-10, where $\psi_s = f(x, y_s)$ represents the profit objective function in scenario s; x is the decision variable, y_s is the control variable in scenario s; λ is the weighted solution error. The larger λ is, the lower the solution is sensitivity to changes in the external environment.

$$\sigma = \sum_{s\in\Omega} P_s\psi_s + \lambda \sum_{s\in\Omega} P_s[(\psi_s - \sum_{s\in\Omega} P_s\psi_s) + 2\theta_s] \quad (4\text{-}10)$$

Where,

$$\psi_s - \sum_{s\in\Omega} P_s\psi_s + \theta_s \geq 0 \quad \forall s \in \Omega \quad (4\text{-}11)$$

$$\theta_s \geq 0 \quad \forall s \in \Omega \quad (4\text{-}12)$$

If $\psi_s \geq \sum_{s\in\Omega} P_s\psi_s$, then $\theta_s = 0$; otherwise $\theta_s = \sum_{s\in\Omega} P_s\psi_s - \psi_s$. Then formula 4-6 can be converted to formula 4-13:

$$Min \sum_{s\in\Omega} P_s\psi_s + \lambda \sum_{s\in\Omega} P_s[(\psi_s - \sum_{s\in\Omega} P_s\psi_s) + 2\theta_s] + \omega \sum_{s\in\Omega} P_s\delta_s \quad (4\text{-}13)$$

4.3.1.2 The Production Planning Model for Distributed Manufacturing Systems

The following notation is used in the development of the mathematical model:

Index:

t Time node in the production planning period. T is the set of all time nodes, i.e., $t \in T$;

i,j Material No. of the final product structure tree. J is the set of final products; N is the set of all raw materials. $i.j \in J,N$;

k Plant No. K is the set of all plants, $k \in K$;

s The number of customer order demand scenarios that may appear in plants, $s \in \Omega$.

Parameters:

$M_{i,j}$ The number of children i needed by parent j. If child i has parent j, then $M_{i,j} > 0$; otherwise $M_{i,j} = 0$;

$psc_{j,k}^{s}$ The inventory/stock cost factor for parent j in plant k in scenario s;

$rsc_{i,k}^{s}$ The raw material inventory cost factor for child i in plant k in scenario s;

$D_j^s(t)$ The external demand for parent j in scenario s in period t;

$pc_{j,k}$ The inventory for parent j in plant k;

$rc_{i,k}$ The raw material inventory for child i in plant k;

$B_{i,k}$ The working hours of critical resources needed by child i in plant k;
P_s The possibility of scenario s, $\sum_{s \in \Omega} p_s = 1$;
$C_k(t)$ The available working hours in period t for plant k.

Decision variables:

$P_{j,k}(t)$ The number of processing products in period t for parent j in plant k;
$ST_{j,k}^{s}(t)$ The number in inventory for parent j in plant k in scenario s at the end of period t;
δ_{jt}^{s} The shortfall demand in period t for parent j in scenario s.

The purpose of production planning optimization is to minimize inventory while satisfying the critical capacity constraints and customer requirements. Therefore, the inventory/shortage cost factor is introduced in the optimization process. The objective function to minimize the product shortage cost and inventory cost for distributed manufacturing systems is presented as follows:

$$\min\ Z = \sum_{k \in K} \sum_{i \in N} \sum_{t \in T} \left[psk_{j,k}^{s} \cdot \left(\delta_{jt}^{s} + ST_{i,k}^{s}(t) \right) + rsc_{i,k}^{s} \cdot ST_{i,k}^{s}(t) \right] \tag{4-14}$$

s.t.

$$ST_{i,k}^{s}(t-1) + P_{i,k}(t) - ST_{i,k}^{s}(t) + \delta_{jt}^{s} = D_{i,k}^{s}(t) + M_{i,f(i)}(t) P_{f(i),k}(t) \quad i \in J, t \in T, \forall s \in \Omega \tag{4-15}$$

$$ST_{i,k}^{s}(t-1) + P_{i,k}(t) - ST_{i,k}^{s}(t) = M_{i,f(i)}(t) P_{f(i),k}(t)$$

$$i \in N - J, t \in T, \forall s \in \Omega \tag{4-16}$$

$$\sum_{i \in N} B_{i,k} P_{i,k}(t) \le C_k(t)$$

$$k \in K, t \in T, \forall s \in \Omega \tag{4-17}$$

$$P_{i,k}(t), C_k(t) \ge 0 \ \ pc_{j,k} \ge ST_{j,k}^{s}(t), rc_{i,k} \ge ST_{j,k}^{s}(t)$$

$$i \in N, t \in T, \forall s \in \Omega \tag{4-18}$$

Formula 4-14 is the objective function, which is to minimize the total cost of the product shortage and inventory. Formula 4-15 is the equilibrium equation to express the material demand of the final products that customers needed. Formula 4-16 is the equilibrium equation to express the material demand of components. Formula 4-17 represents the demand of critical capacities. Formula 4-18 indicates that the number of products or components produced in every period cannot be negative.

4.3.1.3 Robust Production Planning Optimization Model for Distributed Manufacturing Systems

The most important step to introduce uncertain factors at the production planning stage is to develop a model to describe uncertainties. In terms of the basic principles of the robust optimization model, a robust optimization model is developed according to equation 4-13. As shown in formula 4-19, the first part is the robust function of solutions, and the second part is the robustness of model.

$$Min\left(\sum_{s\in\Omega} P_s z_s + \lambda \sum_{s\in\Omega} P_s[(z_s - \sum_{s\in\Omega} P_s z_s) + 2\theta_s]\right) + \omega \sum_{s\in\Omega} P_s \delta_{jt}^{s} \tag{4-19}$$

S.t.

$$z_s - \sum_{s\in\Omega} P_s z_s + z\theta_s \geq 0 \quad \forall s \in \Omega \tag{4-20}$$

$$z\theta_s \geq 0 \quad \forall s \in \Omega \tag{4-21}$$

4.3.2 Production Planning in MASs

The Agent-based push-pull production planning and control system structure that was developed in section 3.4 is used to satisfy requirements of production management in the distributed network environment. In this structure, a production plan is completed jointly by using a cooperative planning Agent, a resource management Agent and an order/product demand management Agent. The production planning process in distributed manufacturing systems consists of two progressive stages: one is production planning and capacity balance of final products; another is production planning and capacity balance of components. Hence, the internal structure of a Multi-Agent production planning system for distributed manufacturing systems could be further refined to construct a Multi-Agent hierarchical production planning system as shown in Figure 4-4.

The basic feature of distributed manufacturing systems has determined that their production planning mode is a distributed, group decision-making process. The traditional production planning mode is a centralized decision-making process for internal production activities of a single manufacturing system. Most modern enterprises are a distributed system consisting of multiple manufacturing systems; each system has a separate legal entity and its own core decision-making ability. The production plan of each member in distributed manufacturing systems is affected by other members' production decisions, it is necessary to collaborate among members. The distributed manufacturing production planning process is a collaboration and communication process. In a multi-layer coordination structure, a Multi-Agent-based production planning system has the following elements in common:

1) *Multi-layer collaboration*: Agents in each layer complete the task demands of a parent Agent in a distributed collaborative way by sharing knowledge and information in order to achieve local control. In particular, the cooperative planning Agent in each layer receives the demand information of a parent cooperative planning Agent, and then decomposes it into production planning processes in a certain sequence, finally allocates them to lower-layer Agents in turn, (i.e., a multi-layer collaborative planning process).
2) *Distributed collaboration:* A single Agent has its own intelligence so that it can optimize the process locally. Therefore, it is possible for Agents in the same layer to interact with each other, which can be regarded as a local optimization strategy. Similar to the collaboration-based approach presented previously, the collaboration style of Agents in the same layer is determined by the information-sharing degree of Agents in the same layer. In the case of complete information sharing, a MAS collaborates by using the collaboration agreement. In the case of incomplete information sharing, a MAS collaborates by using the negotiation agreement while considering the competitive relationship between enterprises.

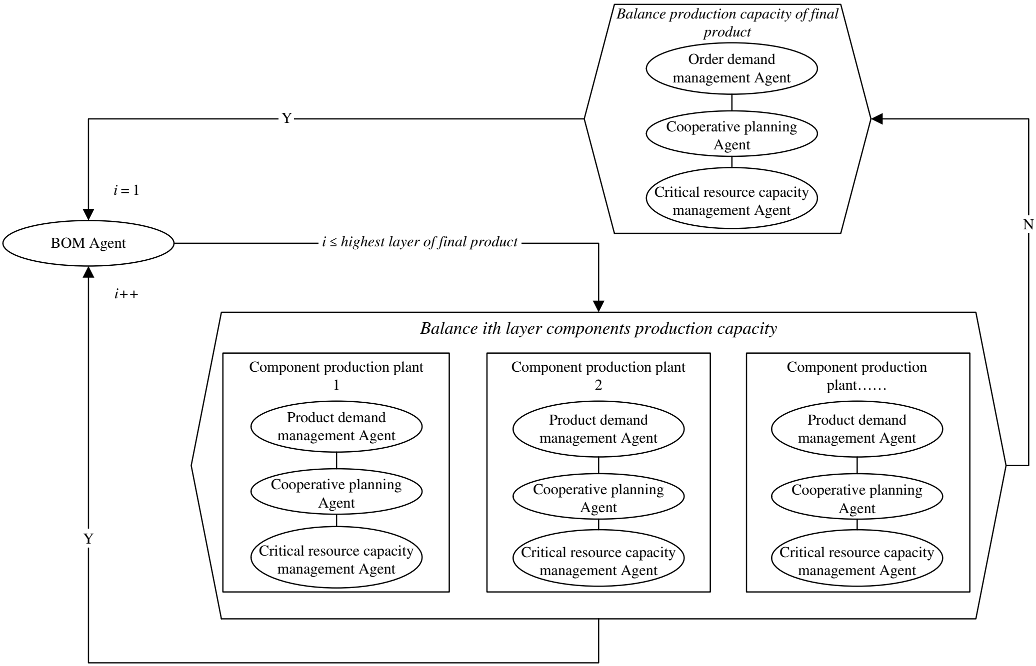

Figure 4-4 The structure of a Multi-Agent production planning system for distributed manufacturing systems.

The advantage of the MAS-based production planning optimization method for distributed manufacturing systems is that it decomposes production planning problems vertically and horizontally in order to integrate related enterprises in time and space. Its optimization objective is to manufacture best products with the least cost in the shortest time so as to satisfy customers' personalized demands. Since this method integrates multiple Agents to make decisions, the decision-making process has some features of autonomy and complexity. The production plan determines what to do according to the ability of an enterprise. It improves the customer service level, inventory turn and productivity, and updates and sustains the feasibility and effectiveness of the plan by proportionately arranging the production process. Besides, it makes sure that the maximum available amount of material and the maximum capacity of each machine have not been exceeded. More specifically, it should follow some basic principles:

1) *Least product types*. Least product type numbers are used to arrange the main production plan. If the number of product types is greater, it will increase the difficulties of prediction and management.
2) *Independence and specification*. The job list only consists of specific types of products or components to be produced. The main production plan should record products or components to be purchased or produced from the perspective of practical demands instead of a plan list. These products are all specific and can be divided into identifiable parts or components.
3) *Critical product types*. Product types that have a significant impact on production capacities, financial indicators or critical materials are recorded. The products that have a great influence on production capacities are those products that affect processing and assembly greatly, for example, some types of mass production products, products that result in bottlenecks, products processed by using critical work centers. As regards financial indicators, these are those products that are most related to the company's profits, for example, products with high manufacturing costs, valuable components and expensive raw materials, or products including some components with special requirements, and products with

relatively inexpensive components being regarded as the main profit source of a company. As regards critical raw materials, these are products that have a very long lead time or limited raw material suppliers.

4) *Comprehensive representation*. Products in the production plan should be as comprehensive as possible to represent the company's processing products. The production plan should cover components as much as possible in order to reflect the information of manufacturing facilities as much as possible, especially bottlenecks or critical work centers.
5) *Appropriate margin*. Appropriate capacity rooms are reserved, and preventive maintenance times are taken into consideration. The preventive maintenance process is arranged as a task in the production plan; the capacities of work centers are reduced according to preventive maintenance times.
6) *Basic stability*. The production plan should remain stable over a certain period. The approach that changes the production plan randomly according to subjective desire will damage the original reasonable normal limited plan and weaken system planning capability.

4.3.3 The Running Model of a Multi-Agent-Based Production Planning System

In order to optimize the production plan for distributed manufacturing systems, the running model of a Multi-Agent-based production planning system is developed in Figure 4-5.

1) The order/product demand management Agent develops a production plan in accordance with the aggregated and sorted customer orders, which include both products in a stable life cycle and those that have never been produced before. The former class is taken into consideration in this book. The production plan is developed by considering beginning inventory, current demand, specific planned amount, and time factors in combination with operation sequences and resource constraints.
2) The production plan is arranged for plants to manufacture final products. The future market demand for products in a mature and stable life cycle is predictable. According to the orders received in the first step, the cooperative planning Agent

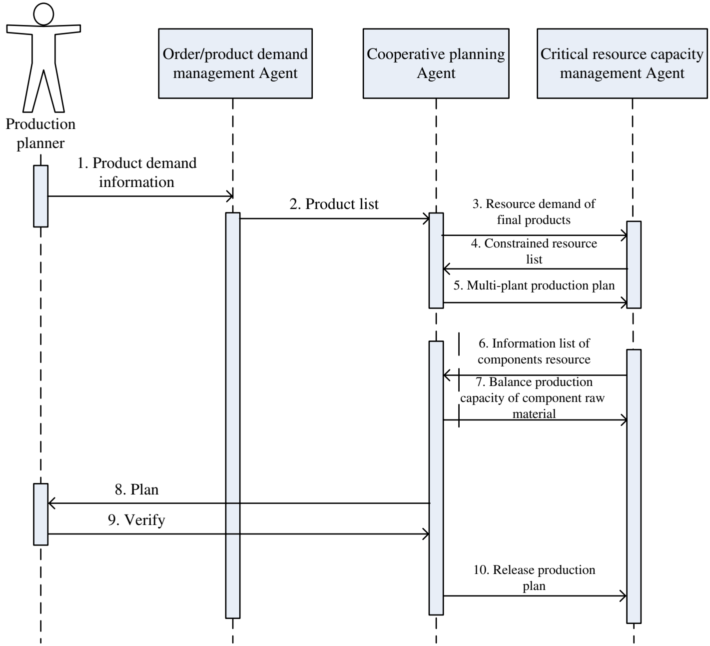

Figure 4-5 The running model of a Multi-Agent-based production planning system.

balances capacities of plants and arranges a pre-production plan according to the resource capacity information provided by the critical resource capacity management Agent. Then the pre-production plan is obtained by the optimization method, and the pre-production plan is formulated by combining data from the first step and the second step.

3) The production plan of the raw material supply factory is developed. The cooperative planning Agent considers the arranged plan in the previous layer to be the production planning constraints in the current layer. This allows the Agent to obtain the amount of component resources needed for the final products. Then it can balance resource capabilities of distributed plants, which are provided by the critical resource capacity management Agent, in order to form the production plan for the raw material supply plant, to control the supply speed of each component of final products.

4.4 Agents in Multi-Agent Production Planning Systems

The interaction process among Agents during the execution process of the production plan is determined by the structure of a Multi-Agent production planning system. In order to provide details of each Agent's behavior and performance, the following section presents structures, behavior models, functions and running modes of an order/product demand management Agent, a cooperative planning Agent and a critical resource capacity management Agent in Multi-Agent production planning systems.

4.4.1 Order Demand Management Agent

As shown in Figure 4-6, an order/product demand management Agent is adopted to receive customer orders, to exchange information with upstream and downstream customers and suppliers just in time, to submit order execution progress to customers, and to release components tasks to corresponding plants in time. Moreover, the order/product demand management Agent is used to release the order demand to a cooperative planning

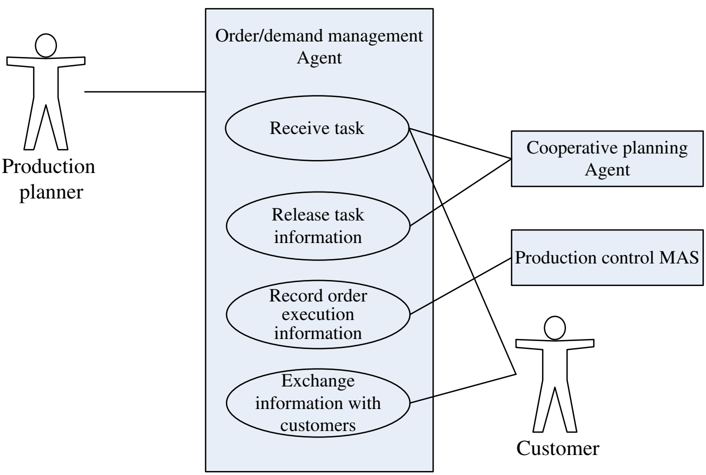

Figure 4-6 The use case diagram of an order/product demand management Agent.

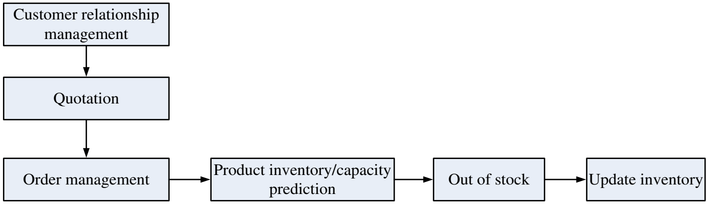

Figure 4-7 The management flow of the order/product demand management Agent.

Agent so as to formulate a production plan and to balance critical resource capacities.

As shown in Figure 4-7, The order/product demand management Agent is employed to manage the product sale contract of clients and manufacturing systems, to carry out communications between customers and enterprises, and to pass the information to sales in the internal system. The order/product demand management Agent is adapted to control sales in a planned manner by maintaining and managing customer orders in order to maintain sale activities, production activities and purchase activities in an orderly, smooth and efficient manner. The process flow of the order/product demand management Agent is presented as follows:

1) Sale orders are uploaded manually or generated by a quotation. The quotation can be set to confirm whether it is to be considered in the calculation range of an MRP.
2) While sale orders are uploading, the sales contract–related information, such as order release dates and due dates, is recorded simultaneously.
3) An order number is generated automatically.
4) While products needed by customers are uploading, the system will calculate the promising quantity of products/promising capacity information in accordance with the existing inventory and capacity.
5) The order-created dates, creators, last modified dates and modifiers are recorded so that late information can be traced.
6) When the order information is changed, the order change version information is recorded in order to facilitate the latter query.

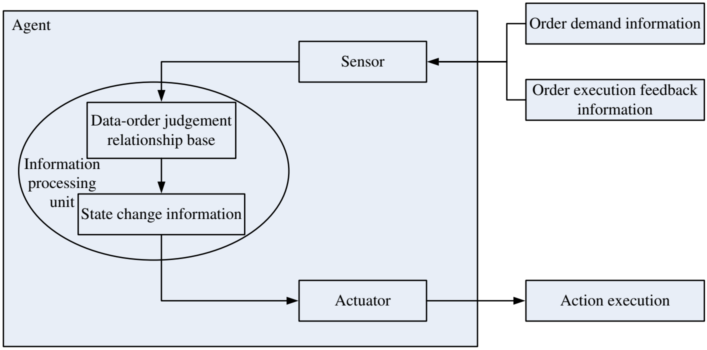

Figure 4-8 The mechanism of the order/product demand management Agent.

The order/product demand management Agent is used to manage customers in relationship to orders, and to provide real-time order status information for customers. During the entire process, the order/product demand management Agent provides a variety of real-time feedback information according to the customer order information. The order/product demand management Agent is a kind of reactive Agent. The structure of the order/product demand management Agent is shown in Figure 4-8.

4.4.2 Cooperative Planning Agent

A cooperative planning Agent is used to develop a production plan and to balance capacities. In distributed manufacturing systems, orders/products are allocated according to the spare capacities of each plant to determine the amount of orders/products for each plant. In the production planning process, according to the critical resource capacities of plants, the tasks released by the order/product demand management Agent are assigned to different plants, in different periods, in terms of certain rules and objectives. The behavior model of the cooperative planning Agent is shown in Figure 4-9. According to the information-sharing degree of distributed plants, a negotiation protocol or a collaboration protocol is used to arrange the production plan.

The cooperative planning Agent is adapted to formulate the prediction plan by accepting orders and conducting sale

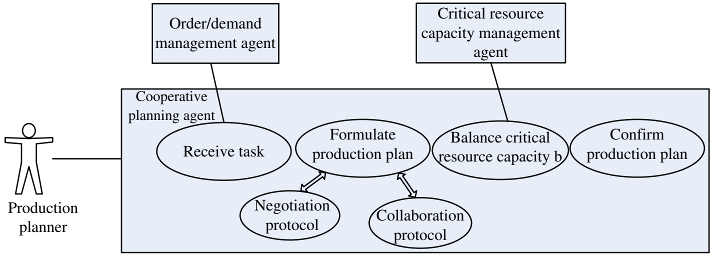

Figure 4-9 The use case diagram of a cooperative planning Agent.

predictions, and to trigger the production planning process by regarding the prediction results and customer orders as demands. The production plan is calculated by arranging tasks on the basis of information, such as BOM configuration processing routes and inventory, in order to obtain a plan, which not only meets the total market demand, but also is relatively stable and balanced. This is the basis for a manufacturing enterprise to maintain sustainable development.

The running process of the cooperative planning Agent involves a great deal of basic information, parameter setting and complex parameters-handling. Hence, the cooperative planning Agent is a kind of thinking Agent. Its structure is illustrated in Figure 4-10. In the information processing process of this structure, an information fusion model corresponds to a planning model; a knowledge base corresponds to production planning optimization methods; objectives correspond to various performance indicators. Finally, a production plan and a purchase plan are generated to guide the manufacturing execution process.

4.4.3 Critical Resource Capacity Management Agent

A critical resource capacity management Agent is used to balance production capacities in the production planning process. When the number of tasks assigned by the cooperative planning Agent has exceeded the maximum capacity of critical resources in a plant, the critical resource capacity management Agent should deliver this information to the cooperative planning Agent in order to trigger the cooperative planning Agent to change the original production plan.

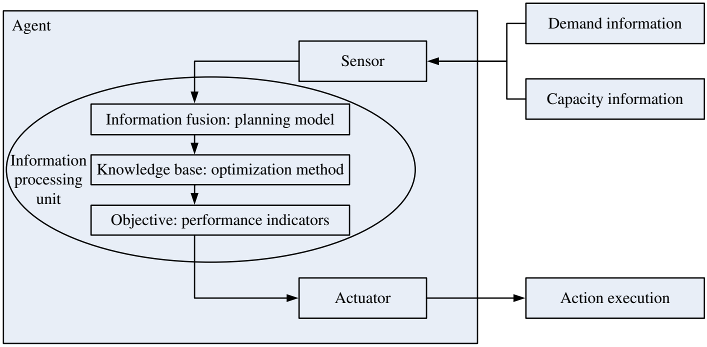

Figure 4-10 The mechanism of a cooperative planning Agent.

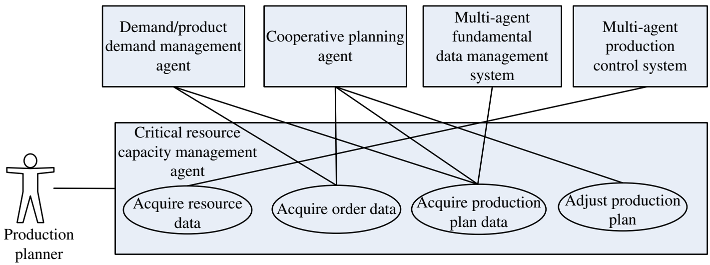

Figure 4-11 The use case diagram of a critical resource capacity management Agent.

Figure 4-11 shows the main functions of the critical resource capacity management Agent. Sophisticated critical resource management can not only greatly improve the efficiency of manufacturing systems and resource utilization, reduce operating costs, facilitate various activities of manufacturing systems, but also discover a variety of problems that appear in the running process, and then improve it.

The critical resource capacity management Agent provides the feedback of production planning results, balances critical resource capacities and judge whether planning results satisfy capacities of critical resources. Hence, the critical resource capacity management Agent is a kind of reactive Agent. The structure of the critical resource capacity management Agent is shown in Figure 4-12.

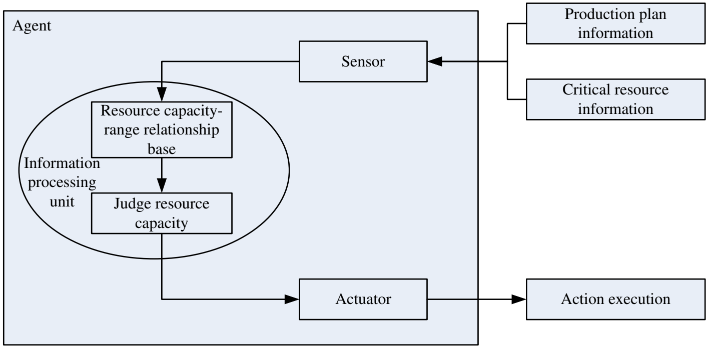

Figure 4-12 The mechanism of a critical resource capacity management Agent.

4.5 Contract Net Protocol-Based Production Planning Optimization Method

In view of information sharing among multiple plants in distributed manufacturing systems, the cooperative production planning process among plants must be taken into consideration, to manufacture appropriate products at the lowest cost and with the fastest speed in order to satisfy customers' demands. That is to say, the Agents in the MAS are required to collaborate with each other so as to reach a common goal, which conforms to basic standards of collaborative protocols. Since the contract net protocol is the most popular one among collaborative protocols for Multi-Agent Systems, the contract net protocol–based production planning algorithm for distributed manufacturing systems is given in the following section.

4.5.1 Contract Net Protocol

In 1980, Smith[26] proposed a contract net protocol to solve distributed problems. As a task allocation and cooperation mechanism for negotiation, the contract net protocol simulates the tendering-bidding-bidding-winner mechanism in market behavior. The bidding values are regarded as task assignment rules, the collaboration and competition among Agents are adopted to solve dynamic, distributed and adaptive task assignment problems.

The collaboration in the contract net protocol has the following characteristics: 1) collaboration is a local process that does not involve the central control; 2) a two-way exchange of information; 3) two sides of the negotiation evaluate the information according to their respective values; 4) the final contract is obtained by a two-way choice. Therefore, this collaboration is entirely dependent on the self-decision and control strategy at every solving node, and the adaptive task assignment is accomplished in a distributed control manner. The actual contract net protocol provides a contract agreement and defines the task assignment, related Agents, and the implementation process.

In the contract net protocol, Agents are classified into two categories: executive Agents and management Agents. The procedure for implementing the contract net protocol-based MAS is outlined clearly in Figure 4-13.

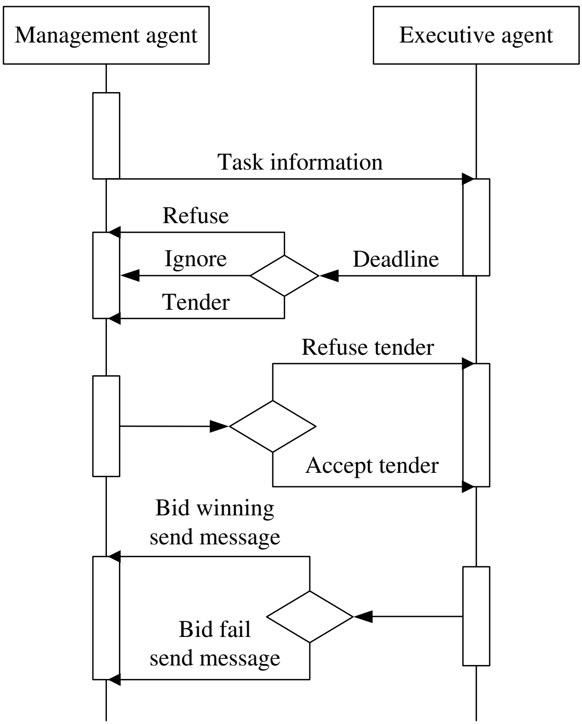

Figure 4-13 The procedure for implementing the contract net protocol.

4.5.1.1 Initialization

Each executive Agent in the system must report its location, status and ability and other information to the management Agent. The purpose is to reduce unnecessary communication in subsequent steps.

4.5.1.2 Task Release

When a waiting task arrives at the management Agent, the Agent decomposes it into sub-tasks and generates a tendering deadline on every sub-task, then sends task tendering information to its corresponding executive Agent.

4.5.1.3 Tendering

When the tendering information is received, every executive Agent evaluates the information according to its state and knowledge base. If the evaluation result is "satisfactory", then the executive Agent will send the bidding information to the management Agent.

4.5.1.4 Issuing

When the tendering deadline arrives, the management Agent will select the best bidding value. It will send "award" information about the bidding winner to the selected executive Agent, and send "failed" information to other executive Agents.

4.5.1.5 Task Execution

When the executive Agent receives the "award" information, it will add this task in the task list. This task will be accomplished at the appropriate time, and then the executive Agent will return the execution information.

An important function of the contract net protocol based on the production planning MAS is to allocate orders/products in manufacturing systems by Agent collaboration. In distributed manufacturing systems, various Agents, such as order/product demand management Agents, cooperative planning Agents, critical resource capacity management Agents, and so on, work collaboratively to formulate the production plan. The critical resource capacity information of plants is collected by a critical resource capacity management Agent, and then a cooperative planning Agent is used to formulate the production plan.

Figure 4-14 shows the hierarchical decision model of the production planning process in distributed manufacturing systems. In this model, the order/product demand management Agent is the task manager, which is mainly responsible for receiving and releasing tasks. The contract net protocol works mainly between the order/product demand management Agent and cooperative planning Agents, who act as managers and actuators, respectively. The critical resource capacity management Agent is adopted to provide critical resource capacity information about plants and to generate the bidding information. The cooperative planning Agent is used to determine the number of tasks allocated to each plant in distributed manufacturing systems.

4.5.2 Contract Net Protocol-Based Collaborative Production Planning Algorithm

The production planning process for distributed manufacturing systems is a complex one, which includes the capacity information coordination of cooperative plants and the task assignment among them. The information exchange process among cooperative plants can further be divided into several stages; different Agents are used in different stages in order to disperse the load of negotiation and adapt to different task assignment requirements of different Agents.

The order/product task assignment negotiation process consists of three stages: 1) The order/product demand management Agent sends the bidding notice about the order/product tasks to cooperative planning Agents; 2) The cooperative planning Agents start to bid in accordance with the capacity information provided by the critical resource capacity management Agent; 3) The order/product demand management Agent audits the bidding information provided by cooperative planning Agents according to content and time requirements of service, in order to determine the bidding winner and send the winning message to the bidding winner. Such a process formulates the collaboration mechanism of cooperative planning Agents in task-driven distributed manufacturing systems.

The contract net protocol-based production planning process is completed by the internal coordination of cooperative planning Agents. The internal coordination is the process that cooperative

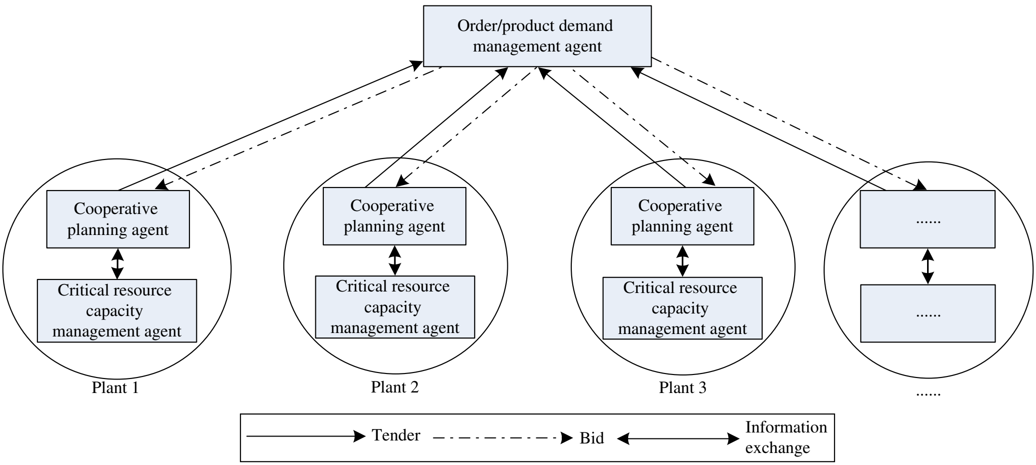

Figure 4-14 Structure of the contract net protocol-based production planning MAS.

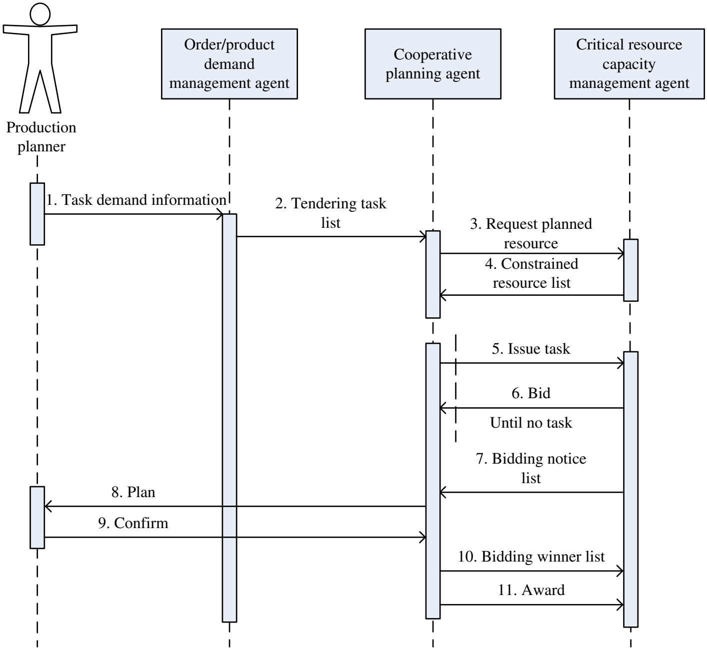

Figure 4-15 The collaborative process of the contract net protocol-based production planning MAS.

planning Agents use to collaborate resource arrangement according to the bidding information provided by critical resource capacity management Agents. The production planning process is the task tendering and bidding process between the order/product demand management Agent and cooperative planning Agents, and the production capacity balancing process between cooperative planning Agents and multi-plant critical resource capacity management Agents. Figure 4-15 illustrates the interactive information in the collaborative protocol-based production planning process. This diagram also indicates the contract net protocol-based order/product task assignment negotiation process.

4.5.2.1 Tendering

The order/product demand management Agent sends out bidding notices about task assignment according to the task list.

The cooperative planning Agent is used to determine the amount of critical resources each plant needs, and then to send the tendering information of each component and product to the designated plant in order.

4.5.2.2 Bidding

After the tendering information is received, critical resource capacity management Agents of plants determine their own tendering value, which is mainly dependent on three indicators:

1) While considering uncertainties, the possible processing time arrangement and completion time are calculated as the tendering value of plants according to the plants' critical resource capacity parameters. Moreover, the predictive task delay cost is given by critical resource capacity management Agents of the plants.
2) The earliest completion times of tasks are provided by critical resource capacity management Agents. The proper resource idle time is determined by searching tasks processed to be processed by critical resource capacity management Agents in order to predict the completion time.
3) The queue length of all the tasks processed by the plant.

Then, the critical resource capacity management Agent sends the bidding information to the cooperative planning Agent. The bidding information contains the three indicators mentioned above.

4.5.2.3 Signing

After the cooperative planning Agent receives the bidding information provided by the critical resource capacity management Agents, the cooperative planning Agent will evaluate the bidding information according to the price and the queue length of all the tasks to be processed by using critical resource capacity management Agents. The contracting rules are presented as follows:

1) Obtain scenario requirements and parameters of tasks provided by critical resource capacity management Agents, and generate the scenario feature list of each plant.
2) Formula 4-19 is used to optimize to obtain the number of tasks processed in each plant.

3) Send out the contract signing information. Plants receive the order/product tasks and add them to the queue, to support production scheduling of plants.

4.5.2.4 Implementation

In the contract net protocol-based production planning MAS, the order/product demand management Agent unites the critical resource capacity management Agent of each plant by using a centralized control structure to work together and provide decision support for all members. The cooperative planning Agent is related to highly cohesive plants or enterprises in distributed manufacturing systems. The critical resource capacity management Agent can obtain the plan or equipment capacity and inventory data in manufacturing systems in any way. This kind of collaboration process among Agents can improve complementarity among individual plans and enhance the effectiveness of the global production plan.

4.5.3 Case Study

In this section, practical data collected from a distributed manufacturing enterprise is used to demonstrate the effectiveness of the contract net protocol-based production planning method. The distributed manufacturing enterprise consists of three plants, which work together to manufacture product 1. The BOM of product 1 is shown in Table 4-2. Eight kinds of raw material are supplied by a number of factories. The supply capability is listed in Table 4-6. The product inventory cost, component inventory cost, inventory capacity and processing time are summarized in Table 4-3. Table 4-4 shows demands in different scenarios. The capacity of plants is shown in Table 4-5. Scenarios are generated

Table 4-2 BOM of product 1.

	Raw material							
product	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	2	3	2	4	4	5	1	2

Table 4-3 Scenario parameters in production planning optimization.

		Product Inventory/ shortage Cost (Yuan)	Raw Material Inventory Cost (Yuan)								Processing Time (h)
Plant	**Scenario**	$J=1$	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	
1	1	5	4	4	4	4	5	5	6	6	3.5
	2	6	5	5	5	5	6	6	7	7	
	3	7	6	6	6	6	7	7	8	8	
2	1	8	5	5	5	5	5	5	5	5	3.6
	2	9	6	6	6	6	6	6	6	6	
	3	10	7	7	7	7	7	7	7	7	
3	1	9	5	6	7	8	8	9	9	7	3.3
	2	10	6	7	8	9	9	10	10	8	
	3	11	7	8	9	10	10	11	11	9	
	Initial Product Inventory (Piece)	Product Inventory Capacity (Piece)	Initial Raw Material Inventory (Piece)								Raw Material Inventory Capacity (Piece)
1	2	15	1	2	3	4	5	6	7	8	10000
2	2	10	10	15	20	12	15	20	15	20	12000
3	0	10	10	0	0	10	0	0	10	20	10000

Table 4-4 Demand of Product 1.

	Time node (t)								
Scenario	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	100	250	350	300	100	200	250	0	100
2	200	250	300	350	200	200	200	350	400
3	150	200	250	300	100	50	0	100	200

Table 4-5 Available production capacity.

Plant	Available Production Capacity (Available Working Hours per Time Node)								
	1	2	3	4	5	6	7	8	9
1	144	160	168	176	120	192	200	200	192
2	144	160	168	176	120	192	200	200	192
3	144	160	168	176	120	192	200	200	192

Table 4-6 Available Amount of Raw Material per Time Node.

1	2	3	4	5	6	7	8
3500	2500	3500	3500	3500	3500	2500	4000

by using stochastic simulation methods,[27] and the final product demand is satisfied with the uniform distribution of a certain interval. The occurrence probabilities of three scenarios are 0.3, 0.3 and 0.4. Parameter ω is 1, which reflects the robustness of the model.

After balance calculation in different layers, the results of production planning process are presented in Table 4-7 and Table 4-8. According to the method introduced above, each plant takes part in the bidding process. After receiving the quote from the cooperative planning Agent of each plant, the cooperative planning Agent in the upper layer compares the time and cost of each critical resource capacity management Agent and sends out the bidding winner information. Then the plan is formulated through the contract net protocol-based collaboration process. The optimal solutions in the situation with nondeterministic demand are compared with the optimal solutions in the situation with deterministic average demand. The results reveals that: 1) The increment of the total cost in a nondeterministic environment is 7.67%, which indicates that the robust optimization results are relatively conservative; 2) the decrement of the standard deviation of the total cost obtained by using the robust optimization strategy is 20.39%, which indicates that the model

Table 4-7 Production Plan of the Contract Net Protocol-Based Collaboration MAS in a Nondeterministic Environment.

	Time node (*t*)								
Plant	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	20	59	38					10	52
2	20	20	50	53	30	35	20	30	62
3	23		46	59	20	32	26		
Inventory/shortage cost: 4603									
Standard deviation: 10.46									

Table 4-8 production plan of the contract net protocol-based collaboration MAS in a deterministic environment.

	Time Node (*t*)								
Plant	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	20	30	38			80	75	65	52
2	24	82		53	33	35	20	30	62
3	31		39	38	82		53		
Total cost: 4250									
Standard deviation: 13.14									

is robust, that is, the robust optimization strategy obtains better robustness at higher cost.

4.6 Bid Auction Protocol-Based Production Planning Optimization Method

In view of the situation of incomplete information sharing among multiple plants in distributed manufacturing systems, Agents are used to compete to make a union-decision in the production planning process, and to optimize globally production in each plant and the whole system, which are basic norms of the negotiation protocol. The bid auction protocol is the most popular one among negotiation protocols in MASs. Therefore, the bid auction protocol-based production planning method is developed for solving Multi-Agent negotiation protocol problems in distributed manufacturing systems with incomplete information sharing.

4.6.1 Bid Auction Protocol

The bid auction is a common procedure throughout human societies. When people want to decide the final value of an item, the auction is a common method, in which the seller will get the most benefit. The typical auction rules are Dutch auction (Auction-Dutch) that originated in the Netherlands, and the English auction (Auction-English) that originated in England.[28] The Auction-Dutch is a price-cutting auction in which the auctioneer first gives an auction price higher than the commodity's value, and then the price will be cut according to certain operating rules until someone is willing to bid or the price hits bottom. The bidder will be the winner of this item. In contrast to Auction-Dutch, Auction-English is a price-raising auction in which the auctioneer will raise the price gradually until there is only one bidder. In a distributed environment, both Auction-Dutch and Auction-English lead to repeated and redundant communication.

In the simple bid auction transaction mode, sellers bid in a buyer's market; the seller who offers the lowest price will sell his item. Buyers bid in a seller's market, the buyer who offers the highest price will get the item. It is noted that this kind of transaction mode will reduce repeated communication. However, it is necessary to reason dynamically concerning the wishes of other sellers and buyers to obtain the true value of a commodity. In this transaction mode, there may be some selfish individuals who offer a false price, for the sake of their own improper benefits, which results in a false deal.

A new bid auction strategy that selects the bidder offering the second-highest price is has been proposed to encourage buyers and sellers to bid in accordance with their wishes and avoid speculation in the transaction process. Although the winner is still the bidder who offers the highest or lowest price, the final transaction price between sellers and buyers is the second-highest price, or the next-lowest price, but not the highest or lowest price. For example, there is an auction of goods, and there are a number of buyers to take part in the bid auction. If bidder A is willing to buy the good with X dollars, and prices that other bidders offer do not exceed X, then the good will be sold to A with a price lower than X, that is, the highest price that other bidders offer. This kind of bid auction mode can also reduce complexity and redundancy of communication as the simple bid auction transaction mode. Besides, it can avoid the situation where bidders cannot offer prices in accordance with their wishes, which increases the possibility and honesty of transaction.

In view of incomplete information sharing among plants, the bid auction–based negotiation protocol is used to formulate the production plan, in which each plant tries to get the order/product task by highlighting its advantages. In this way, the competition among resources will be more orderly and rational; the whole system will get the most benefit; and each task will be the completed by the most appropriate and qualified resource. The order/product tasks will be transported to the cooperative planning Agent by their corresponding order/product demand management Agents, and will be auctioned by critical resource capacity management Agents among all the plants that have available capacities. Within a given period, the tasks will be assigned to cooperative planning Agents of the plants that offer the best price in order to obtain the solution with a most proper execution resource.

4.6.2 The Bid Auction Protocol-Based Negotiating Production Planning Algorithm

The bid auction process is carried out by the negotiation of multiple Agents to locally optimize Agents within a competitive relationship. In the Multi-Agent bid auction negotiation protocol-based production planning system, the negotiation is completed by using the critical resource capacity management Agent, the order/product demand management Agent, the collaborative planning Agent and the BOM Agent. The negotiation process is shown in Figure 4-16.

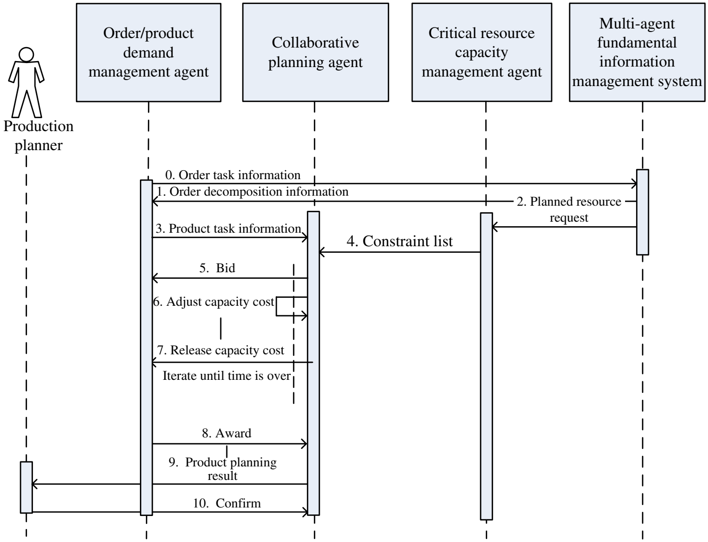

Figure 4-16 Bid auction protocol-based negotiation process in Multi-Agent production planning systems.

The bid auction is adopted to solve Multi-Agent production planning problems, and its purpose is to regard the generation process of a production plan as a bid auction among resources to ensure that tasks can be rationally assigned and completed on time through auction and negotiation.[29] In this mechanism, the task owner is a manager; the resource owner is a signatory; the production task is the item to be auctioned. The resource owner decides the number of bids in accordance with the total quantity of tasks assigned in the enterprise. The task owner selects the resource owner with the lowest cost from all the bidders as signatory in its waiting task list according to the minimum cost rule. And then the task owner sends a signing message to the corresponding plant. The purpose of a production plan is to try its best to satisfy customers' demands. The procedures are presented as follows.

4.6.2.1 Tendering

The order/product demand management Agent sends the bidding notice to tasks in the task set in turn. The cooperative planning Agent determines the amount of critical resources that should be used in each plant, and sends the tendering message to corresponding plants.

4.6.2.2 Auction

After the critical resource capacity management Agent of a plant receives the tendering information, it firstly determines its quote, which is mainly dependent on three indicators:

1) In terms of the critical resource parameters of a plant, the possible processing time arrangement and completion time are calculated as the tender value of a plant while considering uncertainties. The critical resource capacity management Agent of a plant determines the predictive task-delay cost.
2) The critical resource capacity management Agent provides the earliest completion time of a task, determines the proper resource idle time to predict the completion time by searching all the tasks.
3) The queue length of all the tasks processed by the plant.

Then, the critical resource capacity management Agent sends the bidding information to the cooperative planning Agent.

4.6.2.3 Award

In each round of bidding, the task to be signed up and the assigned plant are determined. The order/product planning Agent that issues the tender represents the parent in the product structure tree. Cooperative planning Agents are children in the product structure tree. After the bidding information is provided by critical resource capacity management Agents are received, the cooperative planning Agent judges the bidding information according to the tender value and the queue length of all the tasks processed by critical resource capacity management Agents. The contracting rules are presented as follows:

1) Obtain specific scenario requirements and parameters of completed tasks provided by critical resource capacity management Agents, and generate the scenario feature list of each plant.
2) In the incomplete information-sharing situation, since there is no full-trust relationship between the parent order/product

planning Agent and the child cooperative planning Agent, it is necessary to design the trust factor γ, which indicates that the parent product production plant has to trust the child component manufacturing plant. Assuming that γ is $[0,1]$, then the trust factor γ can be used to evaluate the credibility of the plan proposed by the cooperative planning Agent. $\gamma = 0$ indicates that the parent won't trust the children; and $\gamma = 1$ indicates that the parent will trust the children. In the evaluation process, $\frac{1}{\gamma}$ is used as the weight of cost.

3) Send out the contract signing information. Plants receive order/product tasks and add them into the queue, in order to further support the production scheduling process of plants.

4.6.2.4 Implementation

In the bid auction protocol-based production planning process of distributed manufacturing systems, the protocol is used to assign every task in the waiting task list to a proper production plant and then delete the task from the list. When the protocol ends, the waiting task list is empty. Therefore, the implementation process of this protocol is to conduct the auction-based bidding process iteratively.

4.6.3 Case Study

The case study presented in section 4.6.3 is adopted to illustrate the implementation process of the bid auction protocol. Assume that the information sharing between components production plants and final-product production plants is incomplete, that is, the earliness/tardiness penalty factor is unknown. In addition, the product production plant trusts the components production plant. Here the trust factor vector is $Gamma = [1,0.5,0.6,0.8,0.4,0.5,0.7,0.4]$.

After task assignment and balance calculation, the results of production planning are shown in Table 4-9 and Table 4-10. According to the bid auction negotiation method, each plant takes part in the bidding process. After the cooperative planning Agents of the components production plant give out the quote, the order/product demand management Agent sends out the bidding winner information by comparing all the cost information of distributed plants and considering the trust factors. Since

Table 4-9 Production plan of the bid auction protocol-based collaboration MAS in nondeterministic environment.

	Time node (*t*)								
Plant	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	20	39	53	33	43	20		10	52
2	53	33	45		33	43	20	30	62
3		53	33	45		32	23		
Inventory/shortage cost: 5304									
Standard deviation: 11.02									

Table 4-10 Production plan of the bid auction protocol-based collaboration MAS in deterministic environment.

	Time node (*t*)								
Plant	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	53	33	35		20	30		20	30
2	59	12	32	59	12	32	20	30	62
3	23		46	31		46	26		
Inventory/shortage cost: 4950									
Standard deviation: 13.98									

incomplete information sharing is adopted in distributed plants, the negotiation-based production plan is different from that in the situation of full information sharing.

4.7 Conclusion

This chapter has presented the production planning process for a distributed manufacturing system, which includes two stages: product task assignment and production capacity balance. Besides, a Multi-Agent production planning system has been proposed to solve production planning problems in distributed manufacturing systems. With the aid of the collaboration protocol or the negotiation protocol, multiple Agents in this MAS system formulate effective production plans. When plants collaborate in a situation with complete information sharing, the contract net protocol is used to implement the collaboration-based production planning process. When plants collaborate in a situation with incomplete information sharing, the bid auction protocol is adopted to implement the negotiation-based production planning process. In this production planning process for distributed manufacturing systems, the actual production process has been taken into consideration, and a case study has been included to demonstrate the effectiveness of the proposed method.

References

[1] Renw, W., Nectarios, K., Rodolfo, G., et al. Using Agent for Solving a Multi-Commodity-Flow Problem. *European Journal of Operational Research*, 2009, 194(3): 888–900

[2] Garey, M. R., and Johnson, D. S. *Computers and Intractability: A Guide to the Theory of NP-completeness*. San Francisco: W. H. Freeman, 1979.

[3] Aarrdal, K., and Larsson, T. A Benders Decomposition-Based Heuristic for the Hierarchical Production Planning Problem. *European Journal of Operational Research*, 1990, 45(1): 4–14.

[4] Ashour, S., and Hiremath, S. R. A Branch and Bound Approach to the Job Shop Scheduling Problem. *International Journal of Production Research*, 1973, 11 (1): 47–48.

[5] Wang Chaohui, Chen Haoxuan, and Hu Baosheng. A Lagrangian Relaxation-Based Method for the Scheduling Problem with Batch Process in Chemical Industry. *Control and Decision*, 1997, 12: 408–413.

[6] Qian Songdi. Operations Research. Beijing: Tsinghua University Press, 1990.

[7] Hu Zexin, Shao Huihe, and Jiang Weisun. Modeling and Optimization of Multi-Objective Production Planning for Large-Scale Chemical Industry Enterprises. *Control and Instrument in Chemical Industry*, 1988, 15(4): 29–32.

[8] Hu Yunquan. *Application Case Set of Operations Research*. Beijing: Tsinghua University Press, 1988.
[9] Kondili, E., Pantelides, C. C., and Sargent, R. W. H. A General Algorithm of Rescheduling Batch Operations. *Computer & Chemical Engineering*, 993, 17(2): 211–227.
[10] Sahindis, N. V., and Grossmnna, I. E. MINLP Model for Cyclic Multi-Products Scheduling on Continuous Parallel lines. *Computers Chem. Enging.* 1991, 15(2): 85–103.
[11] Emlaghrbay, S. E. Resource Allocation via Dynamic Programming Activity Network. *European Journal of Operational Research*. 1993, 64(2): 199–215.
[12] Panwalkar, S. S., Iskander, W. A Survey of Scheduling Rules. *Operations Research*, 1977, 25 (1): 45 –61.
[13] Kanakamedala, K. B., Reklaitis, G.V., and Venla, V. Reactive Schedule Modification in Multi-Purpose Batch Chemical Plants. *Industrial Engineering of Chemical Research*, 1994, 33(1): 77–90.
[14] He, D.W., and Kusiak, A. Production Planning and Scheduling in Virtual Manufacturing. *IERC Proceedings of 5th Annual Industrial Engineering Research Conference*. 1996, 491–496
[15] Bussmann, S. An Agent-Oriented Architecture for Holonic Manufacturing Control. *Proceedings of 1st International Workshop on Intelligent Manufacturing Systems, EPFL*. Lausanne, Switzerland, 1998: 1–12.
[16] Dong Hongbin, Wang Jianhua. Research on Multi-Agent Technology. *Computer Application Research*, 1999, 16(10): 29–30.
[17] He Yanxiang. Research on the Model of Multi-Agent Based Distributed Open Computation Environment. *Chinese High Technology Letters*, 1998, 8(6): 16–19.
[18] Zeng Xiaoqing. A Study on Multi-Agent Based Virtual Enterprise Collaborative Production Planning. Central South University, PhD Thesis, 2003.
[19] Li Peigen, Zhang Jie. *The Agility of Reconstruction and Control of the Intelligent Manufacturing System*. Beijing: China Machine Press, 2003, 1.
[20] Lin Tongzeng. *Operations Research*. Beijing: China Machine Press, 1986.
[21] Beale, E. M. L. On Minimizing a Convex Function Subject to Linear Inequalities. *Journal of the Royal Statistical Society, Series B*, 1955, 17(2): 173–184.

[22] Bellman, R., and Zadeh, L. A. Decision Making in a Fuzzy Environment. *Management Science*, 1970, 17(4): 141–161.

[23] Soyer, A. L. Convex Programming with Set-Inclusive Constrained and Applications to Inexact Linear Programming. *Operations Research*, 1973, 21(5): 1154–1157.

[24] Mulvey, J. M., Vanderbei, R. J., and Zenios, S. A. Robust Optimization of Large Scale Systems. *Operations Research*, 1995, 43(2): 264–281.

[25] Yu, C. S., and Li, H. L. A Robust Optimization Model for Stochastic Logistic Problems. *International Journal of Production Economics*, 2000 64 (1–3), 385–397.

[26] Smith, R. G. The Contract Net protocol: High-Level Communication and Control in a Distributed Problem Solve. *Computers, IEEE Transactions*, 1980, C-29(12): 1104–1113.

[27] Diana, N., and Elio, C. Dynamic Portfolio Optimization: Time Decomposition Using the Maximum Principle with a Scenario Approach. *European Journal of Operational Research*, 2005, 163(2): 217–229.

[28] The Foundation for Intelligent Physical Agents, FIPA Personal Tray Travel Assistance Specification. http://www.fipa.org.

[29] Shi Ke and Li Peigen. A Multi-Agent and Contract-Net Based on Scheduling Approach for Agile Manufacturing Cell. *Journal of Huazhong University of Science and Technology*, 2001, 29(7): 32–34.

5

Multi-Agent-Based Production Scheduling for Job Shop Manufacturing Systems

5.1 Introduction

In general, Job Shop manufacturing has been carried out in a discrete manufacturing environment with small to medium batch manufacturing and large product variation. In the manufacturing process of the Job Shop manufacturing system, the products that are formed by materials undergo separate manufacturing operation sequences in various processing resources. In Job Shop manufacturing systems, the production scheduling scheme is regarded as an interface between the production planning layer and the processing resources control layer, which is responsible for arranging material flow and coordinating the production schedule. In the dynamic changeable manufacturing environments, it is imperative that production scheduling be expected to accept the task arrangements of production planning, so as to conduct mid-term job scheduling optimization and to respond to dynamic events in the short term. As pointed out in Chapter 3, the production scheduling scheme is defined as a typical hybrid push-and-pull production process. Scheduling problems in Job Shop manufacturing systems involve many factors; in particular, its running process is full of dynamic events. The complexity of these scheduling problems has led to the recent interest in addressing the problems by using artificial intelligence techniques.[1] Production tasks are assigned by using Multi-Agents in collaboration, based on Agent techniques, in order to simplify the complexity of production scheduling problems, to improve the stability of manufacturing systems, and to reduce management costs.

5.2 Production Scheduling in Job Shop Manufacturing Systems

Production scheduling problems in Job Shop manufacturing systems are characterized as a combinatorial optimization problem subject to highly complex constraints, which belong to the class of non-deterministic polynomial (NP) hard problems. Previous studies have mainly focused on classical scheduling problems in processing operation sectors. In particular, four assumptions are taken into consideration to solve these classical scheduling problems: 1) The set of jobs is deterministic. 2) The operation processing time required in each machine is predefined. 3) All jobs are necessarily available at time zero. That is to say, the release time for jobs is not taken into consideration. 4) There are no dynamic events in Job Shop manufacturing systems. Nevertheless, the actual Job Shop manufacturing system is a complex dynamic system with various dynamic events, for instance, random machine breakdown, preventive machine maintenance, sudden changes in delivery due dates, and so on. There are two important issues with respect to production scheduling scheme. One issue is to accept the task arrangements of production planning. The other issue is to continually update the released production plans in order to rapidly respond to these changes by considering the impact of a dynamic environment. Therefore, how to develop an effective production scheduling scheme is an important means to improve the stability and the efficiency of Job Shop manufacturing systems.

5.2.1 Job Shop Manufacturing Systems

As a functional unit in a distributed manufacturing system, a Job Shop manufacturing system is expected to reasonably organize the production resources in order to complete all the tasks under the production planning phase. A Job Shop manufacturing system consists of a number of manufacturing cells that are generated in two ways. In the former, a manufacturing cell is generated by different types of machines or processes (e.g., an automated flexible manufacturing cell). On the other hand, machines are grouped in terms of their functional similarities.

An automated flexible manufacturing cell involves processing resources, material handling resources and cell controller, and so on.

1) Processing resources
   Processing resources are capable of performing one or more processes. Due to different processing parts, these processing resources can be classified into six classes, such as machining centers, turning centers, computer numerical control (CNC) machines, milling machines, grinders and gear cutting machines, and so on.
2) Material handling resources
   Material handling resources are employed to automatically supply and load/unload parts and fixtures, and to automatically deliver, dispatch and store between processes. These resources contain a variety of conveyors, automatic guided vehicles, industrial robotics, and so on.
3) Cell controller
   In general, a multi-level distributed computer system is adopted as a cell controller to deal with information (e.g., control commands in CNC machine and delivery information in automated material handling systems) in the Job Shop manufacturing system.

Another way to generate manufacturing cells is in terms of their functional similarities, in which machines of a specified type are grouped together. These manufacturing cells include the same types of CNC machines, machining centers and functional cells generated by general machines. In these manufacturing cells, material handling is manual-based without cell controllers.

A Job Shop manufacturing system includes following typical characteristics:

1) Complex manufacturing environment
   In a Job Shop manufacturing system, there are mutual influences and constraints related to jobs, machines, operators, and buffers, and so on. In addition, the impacts of dynamic events are required to be taken into consideration, which cause the complexity of management to be increased.

2) High equipment flexibility
   Due to high flexibility, variability and the ability to manufacture various parts with different processes and to assemble different types of components and products, it is generally believed that a Job Shop manufacturing system satisfies the requirements of low to medium unstable demand with large product variation. In a highly automated manufacturing cell, it is capable of automatically performing processes including manufacturing, assembling, inspecting, handling, storing and so on. Moreover, it is also able to automatically replace parts, tools and fixtures so as to automatically clamp and deliver, and to automatically monitor machining processes.
3) High efficiency
   Job Shop manufacturing is designed to minimize auxiliary time, preparation time and ending time by using an accurate processes-controller so as to achieve high efficiency and improve equipment utilization simultaneously.

5.2.2 Production Scheduling in Job Shop Manufacturing Systems

Usually, production scheduling problems in Job Shop manufacturing systems are characterized as scheduling problems subject to highly complex constraints. These kinds of scheduling problems concern the allocation of limited sources over time to perform tasks among parallel and sequential processes. Each task consists of several processes with constraints related to the operation sequence of each part, the operation processing time required in each machine, and the requirements and the maximum capacities of manufacturing machines. In the manufacturing process of the Job Shop manufacturing system, all the tasks of a specified type undergo their own independent manufacturing operation sequence, and each process occupies a specific processing resource in a manufacturing cell. The aim of the production scheduling scheme is to allocate proper resources to tasks and develop time arrangements in detail, in order to achieve the best production performance while satisfying constraints.

With the development of personalized customer demands, and intensified market competition, in the case of limited resources, as a response to these demands, it is necessary for Job

Shop manufacturing systems to shorten delivery due dates of products, improve the efficiency of the manufacturing system and reduce work-in-process (WIP) inventory. The characteristics of production scheduling problems in Job Shop manufacturing systems are summarized as follows:

1) *Dynamic*
   This refers to the difference between the background based on the production plan and the actual condition in implementing an event.[2] The manufacturing environment in Job Shop manufacturing systems is constantly changing with various dynamic events[3] (e.g., rush orders, random machine breakdown, part rework, and wrong operations), a lot of inaccurate information (e.g., the arrival time of each material, the unit operation processing time required in each machine and the setup time of each machine) and incomplete information (e.g., the quality of materials and the quality of the manufacturing part). The entire manufacturing process cannot be implemented in an orderly way while system parameters change and dynamic events occur.
2) *Multi-constraint*
   This indicates that production scheduling problems are constrained by many conditions. Considering the characteristics of manufacturing process in Job Shop manufacturing systems, there are constraints related to manufacturing processes, and the maximum capacities of manufacturing resources such as machines, operators, tools and auxiliary tools.
3) *Multi-objective*
   This indicates that a variety of different objectives are involved in the procedure of optimization, and these objectives are interrelated and contradictory. In Job Shop manufacturing systems, the optimization objectives include maximizing machine utilization, minimizing makespan and minimizing manufacturing costs.
4) Computational complexity
   This refers to the NP-hard (Non-deterministic Polynomial) feature, which means it is difficult to solve the problem in polynomial time with a number of operations. In Job Shop manufacturing systems, there are four issues with respect to the production scheduling problems, which include

multiple types of equipment, complex collaborative relationships in the manufacturing process, many involved factors, and multiple states. Therefore, production scheduling problems in Job Shop manufacturing systems belong to the class of NP-hard problems due to their computational complexity.

5.2.3 The Related Literature Review

Due to the complexity of production scheduling problems in Job Shop manufacturing systems and the application requirements, numerous approaches have been proposed by modern manufacturing researchers, all of which promote the development of scheduling theory and applications in Job Shop manufacturing systems. During the past few decades, production scheduling problems were a hot topic in the field of Job Shop manufacturing, and many approaches have been developed, which mainly focus on the development of effective production scheduling optimization methods in order to achieve high efficiency and flexibility. These approaches can be briefly classified into five categories as follows:

1) Mathematical programming approaches
   Mathematical programming[4] approaches simplify scheduling problems by formulating a mathematical programming model, and thus solve scheduling optimization problems or approximate optimization problems by using integer programming, dynamic programming and decision analysis methods. Srinoi et al.[5] developed a fuzzy programming model to solve production scheduling problems in a dynamic Job Shop manufacturing environment. In their model, four input variables including machining time, machine priority, delivery due date priority and adjusting time priority were introduced to obtain one output variable that indicates part priority, by which jobs are scheduled. Yu et al.[6] also adopted a similar approach in a flexible manufacturing floor, in which system variables were predefined, and priorities of all objectives were outputted by detecting and inputting changes of these variables. Wan[7] proposed a fuzzy scheduling system to minimize the total weighted tardiness in dynamic Job Shop manufacturing systems.

2) Heuristic approaches

The aim of heuristic approaches[8] is to obtain solutions by simplifying the search process by eliminating irrelevant information about tasks. The search process includes checking the search space, evaluating paths with possible solutions, and recording searched paths. Hu et al.[9] proposed a real-time production scheduling method for Job Shop manufacturing systems. In a dynamic manufacturing environment, considering the occurrence of real-time events, such as some of the completed parts, changes of parts' expiration time and the arrival of rush orders, they established a real-time production scheduling model for Job Shop manufacturing systems based on S-rough sets by studying reselection and rescheduling problems of parts in the real-time scheduling window. Their method, in which the S-rough set theory was successfully applied in the real-time production scheduling field, not only adapted to the dynamic manufacturing environment to reduce the search size of scheduling problems, but also decreased the frequency of rescheduling. Li et al.[10] developed a priority scheduling algorithm based on bottleneck analysis to solve dynamic real-time scheduling problems in manufacturing systems. Moreover, it is common that rule-based scheduling methods in combination with other distribution methods[11–13] are applied in solving Job Shop scheduling problems. Li et al.[14] proposed a genetic algorithm in combination with heuristic rules to determine the best scheduling solution in dynamic scheduling environments. In their method, machine breakdown time was predicted by using mathematical expectation theory. Li et al.[15] designed a real-time Job Shop scheduling algorithm that consists of neural networks and heuristic distribution rules. In this algorithm, offline heuristic rules were obtained with neural networks trained by a genetic algorithm, fuzzy classification of operations was performed according to some dynamic characteristics at the beginning of machining process, and then online scheduling of classified operations was carried out in accordance with heuristic rules.

3) System simulation approaches

Rather than simply pursuing the mathematical model for manufacturing systems, system simulation approaches[16] focus on describing the logic relationship in manufacturing systems so as to analyze results such as all the assignments, sequencing and time selection, and so on. Hence, they are capable of quickly analyzing a given schedule with low cost, and of comparing various solutions in order to choose the optimal scheduling strategy and dynamic system parameters. Shafaei and Brun[17] proposed a rolling time approach to deal with production scheduling problems in the dynamic Job Shop manufacturing environment, in which scheduling rules were introduced to generate scheduling plans, and the generated scheduling plans were compared to one another based on system simulation approaches. Apart from inheriting these ideas, research on how to select scheduling rules has become a hot topic.

4) Artificial intelligence approaches
Production scheduling problem is a combinatorial optimization problem. Hence, since the 1980s, artificial intelligence approaches have been introduced as optimization tools to solve production scheduling problems in Job Shop manufacturing systems. These artificial intelligence approaches include expert systems, neural networks and Multi-Agent systems, and so on. As Fox and Smith[18] pointed out, an expert system was proposed to solve Job Shop scheduling problems. In their method, a knowledge base was developed according to human experts' experiences and analog data. A constraint-oriented inference mechanism was adopted to ensure the consistency of scheduling by using constraint knowledge. Both quantitative knowledge and qualitative knowledge was used to generate complex heuristic rules. Complex data structure information and special technology to manipulate these data were employed. However, it is difficult to establish, evaluate, maintain and upgrade the system, and it may produce not feasible solutions that deviate from optimal and suboptimal solutions. Tang et al.[19] studied the application of neural networks to solve dynamic mix Job Shop scheduling problems, in which the set of the training neural network sample was derived from simulating scheduling rules. Micheal[20] solved real-time flow shop sequencing problems by using neural networks. Mashiko[21] developed a genetic algorithm in combination with neural networks to solve

real-time flow shop scheduling problems. Cowling et al.[22] proposed a Multi-Agent System (MAS) to solve dynamic scheduling problems. In their method, meta-heuristic algorithms were used to obtain the dynamic scheduling strategy, and when to reschedule or repair the scheduling policy was determined by the Agent. Aydin and Öztemel[23] focused on Agent learning and proposed a learning algorithm. Qiao et al.[24] studied the application of Agent technologies in Job Shop scheduling problems and developed a Multi-Agent distributed dynamic Job Shop scheduling scheme based on a contract net protocol bidding mechanism. Shnits et al.[25] suggested a two-level control hierarchy to solve dynamic scheduling problems in flexible manufacturing systems. The higher level was used to determine a dominant decision criterion and relevant scheduling rules based on an analysis of the actual shop status; whilst the lower level used simulation to determine the best scheduling policy to be selected. Kouiss and Pierreval[26] proposed a Multi-Agent architecture in Job Shop manufacturing systems. In this architecture, all the knowledge related to the Agents was stored in the static knowledge layer. Each Agent received the status information in the local scheduling environment and with information from other Agents. Appropriate scheduling rules in the knowledge base were selected to generate production plans.

5) Computational intelligence approaches

Common production scheduling approaches based on computational intelligence in Job Shop manufacturing systems mainly include simulated annealing, fuzzy logic and evolutionary algorithms. For example, two methods[27, 28] established the parallel machines Job Shop scheduling model to minimize makespan, while other studies[29, 30] established the parallel machines Job Shop scheduling model to minimize tardiness and earliness. Gao and Fang[31] studied non-identical parallel multi-machine Job Shop scheduling problems to minimize makespan. In this method, the master-slave network model was designed based on the parallel computing characteristics of genetic algorithms. Huang et al.[32] designed a hybrid genetic algorithm based on a dynamic fitness function to investigate multi-objective, non-identical, parallel machines Job Shop scheduling problems to minimize earliness, tardiness and makespan. Hu et al.[33] proposed an ant colony algorithm–based scheduling method for mixed Job Shop scheduling problems that consists of parallel machines scheduling problems in combination with flow shop scheduling problems. George and Velusamy[34] treated dynamic scheduling problems as several static scheduling problems triggered by dynamic events. When dynamic events happen, the techniques of genetic algorithms will be used to update the whole scheduling to obtain optimal scheduling solutions. However, this method increases computational effort. Pan[35] solved the scheduling problems in a time period as multi-objective optimization problems by using a rolling time window method in combination with genetic algorithms. Liang et al.[35] proposed an offline learning and online scheduling algorithm based on similar ideas. Apart from the above method, there are numerous scheduling approaches based on the rolling time window to divide dynamic scheduling problems into a series of static problems.[37–39]

Indeed, it is difficult to maintain production schedules in an unaltered status in the running procedure of manufacturing systems. The major reason is that the actual manufacturing system confronts many changes: external changes as changing customers' demands to respond rapidly to fluctuating markets, while internal changes such as random machine breakdown, scarce resources and changes in the unit operation processing time. These changes cause the original scheduling performance to be worse or not feasible. Hence, it is necessary to introduce rescheduling or dynamic scheduling.

In addition, a new approach called real-time intelligence has been proposed as an alternative in order to implement production scheduling in dynamic Job Shop manufacturing systems. Real-time intelligence is a new concept that hybridizes real-time technology with artificial intelligence techniques and computational intelligence approaches. Real-time intelligence is mainly applied in the collaborative process of real-time systems, in which the scheduling algorithm is the core as well as a research hotspot. Chen et al.[40] proposed a dynamic scheduling model based on Agent techniques for Job Shop MASs. The collaborative process among Agents is one of the key issues in dynamic scheduling. Determining the collaborative mechanism of dynamic scheduling in MASs is equivalent to solving the joint problems of MASs. In recent years, the positive feedback and negative feedback based on control theory have become a hot topic in the field of dynamic Job Shop manufacturing systems. The positive feedback optimizes the operation performance of manufacturing systems, whilst the negative feedback maintains the stability of their performance. This method is beneficial for manufacturing systems to rapidly respond to random events in dynamic manufacturing environments. Therefore, a highly flexible MAS architecture with double feedbacks will be introduced in this book. This system is able to meet the requirements of Job Shop manufacturing systems in complex dynamic environments. That is to say, it could intelligently, efficiently and quickly respond to customers' demands and dynamic events.

5.3 Multi-Agent Double Feedback–Based Production Scheduling in Job Shop Manufacturing Systems

5.3.1 Principles of Double Feedback Scheduling Strategy

The concept of feedback was proposed by Safonov,[41] Zames[42] and Doyle[43, 44] in the 1980s. Feedback can be briefly classified into two major classes:[45–47] positive feedback and negative feedback. The former increases the impact of the input on the output in order to optimize the specified performance of manufacturing systems; whilst the other one decreases the impact of the input on the output so as to maintain the stability of the system performance. The purpose of feedback is to respond appropriately to objective changes. There are various cases when the feedback method is applied to solve production scheduling problems in manufacturing systems.

In Job Shop manufacturing systems, considering the complex and non-stationary environments with mutation (e.g., changes in production tasks, rush orders, etc.), the negative feedback rescheduling method is adopted to realize the real-time dynamic adjustment in Job Shop manufacturing process so as to stabilize the entire system. Meanwhile, in order to quickly adapt to these mutations, it is necessary to introduce the positive feedback production scheduling method in Job Shop manufacturing systems to facilitate rapid evolution of the system to achieve more optimal states so that the entire manufacturing process can be conducted in an orderly manner. Therefore, a double feedback scheduling strategy that combines the negative feedback rescheduling method with the positive feedback scheduling method is employed in Job Shop manufacturing systems.

In Job Shop manufacturing systems, the positive feedback scheduling process is adopted to complete manufacturing tasks and allocate resources by task decomposition, organization management, and the optimization of the resources allocation. Due to large product variation and large quantities of WIP (Work in Process), the process causes the production scheduling optimization problem to be very complex. In order to improve machine utilization, minimize makespan and reduce manufacturing costs, the positive feedback scheduling optimization method is used to optimize system performance.

Considering various uncertainties in the manufacturing process (e.g., rush orders, random machine breakdown, part-rework and wrong operations), much inaccurate information (e.g., the arrival time of each material, the unit operation processing time required in each machine and the setup time of each machine) and incomplete information (e.g., the quality of materials and the quality of the manufacturing part), Job Shop manufacturing systems are required to effectively, accurately and rapidly respond to these factors in order to maintain the stability of their systems. The negative feedback scheduling method is adopted to achieve rescheduling and maintain the stability of the system to ensure that the entire manufacturing process can be implemented in a dynamic, orderly manner while system parameters change and dynamic events randomly occur.

5.3.2 The Architecture of the Multi-Agent Double Feedback–Based Production Scheduling System

In the case of frequent information communication and various complex information transmission paths in the double feedback scheduling process, the characteristics of Agent technologies (e.g., standard communications and interaction protocol) are adopted to effectively improve the efficiency of information transmission in manufacturing systems. In this book, the Multi-agent double feedback production scheduling in Job Shop manufacturing systems is established based on Agent technologies, shown in Figure 5-1.

In Job Shop manufacturing systems, the positive feedback production scheduling process optimizes the operational performance of manufacturing systems, whilst the negative feedback rescheduling process maintains the stability of their performance. In the architecture of Multi-Agent production planning and control system, the double feedback–based production scheduling process in Job Shop manufacturing systems is to accomplish the tasks and to achieve the control information exchange between the upper production planning layer and the lower production control layer by coordinating the collaborative scheduling Agent, the resource capacity management Agent and the task management Agent in the scheduling layer. In the production scheduling procedure of Job Shop manufacturing systems, the Multi-Agent collaboration is realized by cooperation on various resources to manufacture the same product in organizing production processes. The key collaborative procedure exists in the task allocation procedure of the positive feedback production scheduling and negative feedback rescheduling, in which the collaborative scheduling Agent, the resource capacity management Agent and the task management Agent become interactive to achieve information interaction in the production task allocation procedure by certain interaction protocols, and to ensure that the production plan is properly adjusted while dynamic events randomly occur.

5.3.3 The Running Model for the Multi-Agent Double Feedback–Based Production Scheduling

The double feedback production scheduling procedure in Job Shop manufacturing systems is so complex that it has to be carried out by the Multi-Agent's group decision-making and collaboration, shown in Figure 5-2.

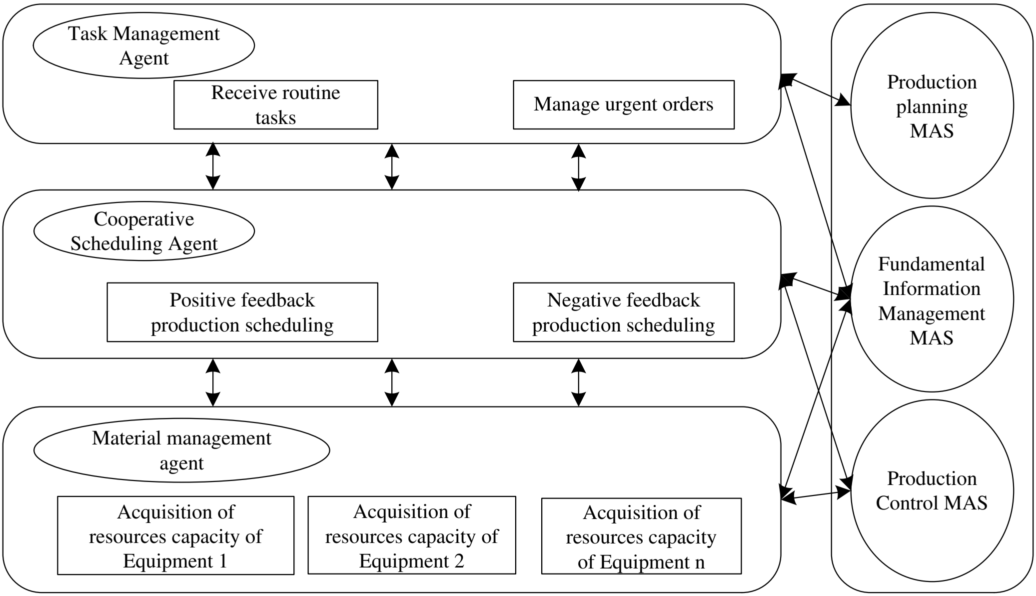

Figure 5-1 The architecture of Multi-agent double feedback based production scheduling in Job Shop manufacturing systems.

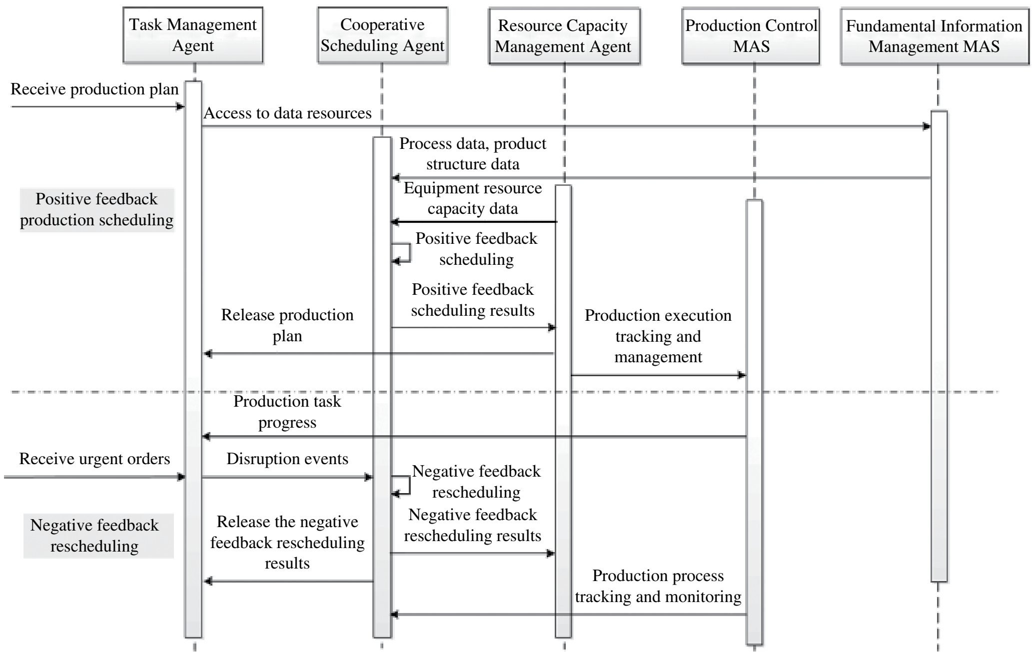

Figure 5-2 Double feedback production scheduling in Job Shop manufacturing systems.

The crucial factors of the double feedback production scheduling are listed below:

5.3.3.1 The Positive Feedback Production Scheduling

Assigned tasks in the production planning phase are obtained by the task management Agent, the resource capacity information is obtained by the resource capacity management Agent, and the process information is obtained by the fundamental information management Agent. When these three kinds of information are aquired, the collaborative scheduling Agent will be used to optimally allocate tasks to machines by designing information exchange protocols among Agents and the task allocation optimization algorithm.

5.3.3.2 The Negative Feedback Rescheduling

Real-time production information on production procedures in Multi-Agent production control systems is obtained by the collaborative scheduling Agent. When dynamic events (e.g., sudden changes in environments and tasks) occur, the collaborative scheduling Agent is employed to choose proper optimization objectives so as to adjust the results of the negative feedback rescheduling process by designing the information exchange protocols among Agents and by developing an optimization algorithm in the negative feedback rescheduling process.

5.4 Agents in the Multi-Agent Double Feedback–Based Scheduling System

Effective production schedules for Job Shop manufacturing systems are formulated by the collaborative scheduling Agent, the resource capacity management Agent and the task management Agent based on the collaboration of the Multi-Agent production control system and the Multi-Agent fundamental information management system. In particular, these two Multi-Agent systems are used to provide basic data and receive tasks. Hence, in this section, we will focus on introducing the architecture and behaviour models of the task management Agent, the collaborative scheduling Agent, and the resource capacity management Agent in Job Shop manufacturing systems.

5.4.1 Task Management Agent

When receiving production tasks assigned by the Multi-Agent production planning system, the task management Agent will be adopted to transform them into a production task list in manufacturing cells. The task management Agent is used to manage a production job list specifying the information for each job – to manufacture a part, for instance – such as the delivery due date, the production volume, the operation sequence of the part, the required materials, and so on. The use case diagram of the task management Agent is illustrated in Figure 5-3. The task management Agent is employed to achieve the real-time task completion circumstances provided by the Multi-Agent production control system, and to feed back the information related to task circumstances to the Multi-Agent production planning layer. When a task is completed, the next operation will start. If the sub-task is the last operation of the corresponding part; then the product will ship to stock.

The task management Agent is adopted to receive tasks assigned by the Multi-Agent production planning system, and to deliver tasks to the collaborative scheduling Agent, so as to collect the production execution information. In the whole procedure of production management, the task management Agent is used to react with the information obtained by sensors. The relationship base in the information-processing unit of the task management Agent mainly comprises options of receive-deliver tasks and design of production-process tracking feedback. Hence, the task management Agent belongs to the reactive Agent, and Figure 5-4 illustrates the mechanism of the task management Agent.

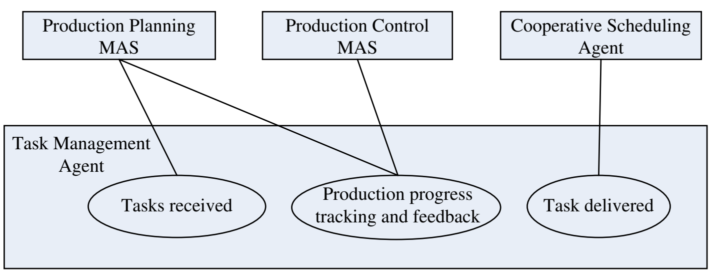

Figure 5-3 The use case diagram of a Task Management Agent.

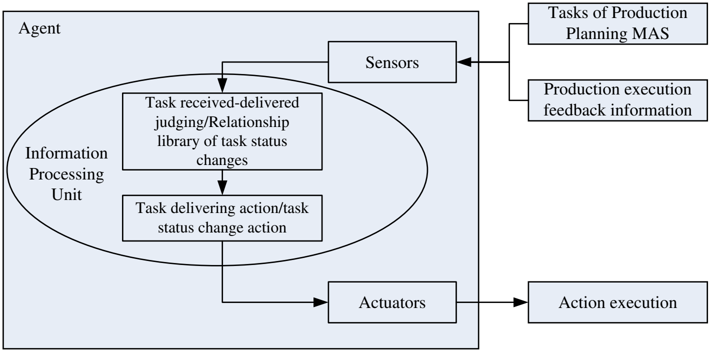

Figure 5-4 The mechanism of the Task Management Agent.

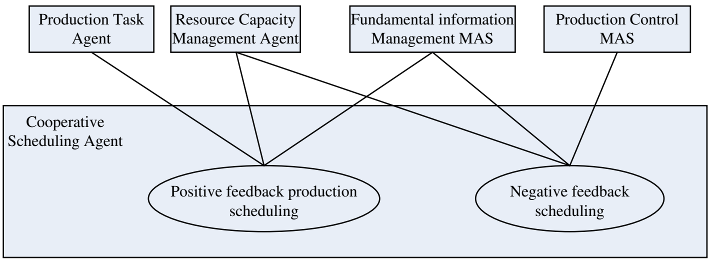

Figure 5-5 The use case diagram of a collaborative scheduling Agent.

5.4.2 Collaborative Scheduling Agent

In Job Shop manufacturing systems, the collaborative scheduling Agent is responsible for two feedback schemes. The positive feedback production scheduling scheme is used to develop production plans, whereas the negative feedback rescheduling scheme is adopted to adjust production plans in accordance with actual circumstances. Certain rules and optimal objectives are proposed to allocate tasks machines and labour resources in the procedure of developing and adjusting production plans. After that, the updated production plans should be delivered to the Multi-Agent production control system to guide the manufacturing process. As shown in Figure 5-5, the behaviour model of the collaborative scheduling Agent is obtained by analysing the application of the collaborative scheduling Agent in Job Shop manufacturing systems.

After receiving production tasks assigned by the Multi-Agent production planning system, the collaborative scheduling Agent is adopted to configure basic information and parameters, and to calculate complex parameters in order to generate production plans satisfying constraints such as manufacturing processes, the delivery due dates of various products, and the maximum capacities of manufacturing resources. The objectives are to minimize the inventory, to maximize the resource utilization incurred in manufacturing the products, and to improve the on-time delivery rate. Since a complex knowledge reasoning process is needed in the production planning phase, the collaborative scheduling Agent is regarded as a typical thinking Agent; its mechanism is presented in Figure 5-6. In the procedure of information processing, the information fusion model represents both the positive feedback scheduling model and the negative feedback rescheduling model, the knowledge base is equivalent to scheduling methods, and the goal corresponds to various performance indicators, and a feasible production plan is generated by optimizing. The principle of information processing unit will be described in detail in section 5.5 and 5.6.

5.4.3 Resource Capacity Management Agent

The resource capacity management Agent is adopted to provide real-time information on resources and to balance capacity so as to achieve production tasks by communicating with the collaborative scheduling Agent, as shown in Figure 5-7. According to

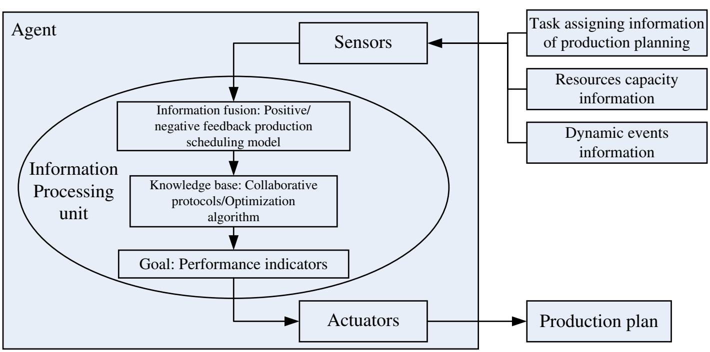

Figure 5-6 The mechanism of the collaborative scheduling Agent.

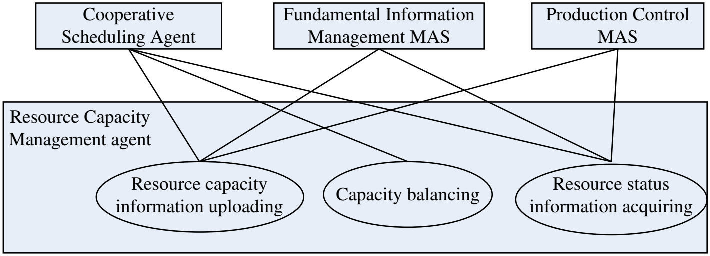

Figure 5-7 The use case diagram of a resource capacity management Agent.

the interaction protocols defined by the collaborative scheduling Agent, the resource capacity management Agent is able to provide a variety of resource capacity data, and to balance capacity based on the results of the positive feedback scheduling or the negative feedback rescheduling obtained by the collaborative scheduling Agent, in order to arrange resources. The real-time changing information of resources is acquired by the resource capacity management Agent, and is reported to the collaborative scheduling Agent.

The resource capacity management Agent is similar to the critical resource capacity management Agent, which is mainly responsible for acquiring resource capacity information, balancing capacity and judging whether the results of production scheduling satisfy the resource capacity constraints. Hence, the resource capacity management Agent belongs to a reactive Agent, and its mechanism is similar to that of the critical resource capacity management Agent presented in section 4.4.3.

5.5 Positive Feedback–Based Production Scheduling in Job Shop Manufacturing Systems

In the positive feedback scheduling phase, the task allocation is completed by using the collaborative scheduling Agent, the task management Agent and the resource capacity management Agent, simultaneously. In this section, according to the specific requirements of the collaborative scheduling Agent in the information-processing phase, the knowledge base that consists of the interaction protocols between Agents and the optimization algorithms is designed in the positive feedback scheduling process. Then the performance indicators are selected and are optimized when the system is operating.

5.5.1 Problem Description

In the production scheduling process for Job Shop manufacturing systems, the task management Agent is used to receive product tasks and to decompose them into process plan tasks, which are received and executed by the resource capacity management Agent. The first crucial step to effectively implement production scheduling is to determine which devices are capable of undertaking these process plan tasks. The following two capacity assessments are considered in the development of production schedules in Job Shop manufacturing systems. First, processing capacity is evaluated. The manufacturing system should be able to provide processing methods and machining precision required to manufacture a part. And the machining machines in the manufacturing system should be able to meet the requirements of product size and processing size. Second, production capacity is judged. In this study, it is important to make sure that the production tasks assigned to the manufacturing system will not exceed its current idle production capacity while still satisfying the delivery due dates. The current idle production capacity of the manufacturing system is calculated by the production plan being executed.

In this case, the following two aspects are taken into consideration to construct feasible production schedules in Job Shop manufacturing systems. The task management Agent is used to minimize the manufacturing costs; whilst the resource capacity management Agent is adopted to improve machine utilization. Therefore, the positive feedback–based production scheduling process in Job Shop manufacturing systems is a two-level process. According to the tasks data and the production capacity, the task management Agent is used to generate the production plan by optimizing the allocation of resources. The production plan generated by the task management Agent is a global plan. The objectives are to minimize total manufacturing costs by globally considering the capacity of each machine, and to meet the customers' demands by quickly manufacturing products. The resource capacity management Agent is adopted to manage each resource unit to minimize makespan.

According to the specific requirements of the task management Agent and the resource capacity management Agent, two-level production scheduling in Job Shop manufacturing systems will be analysed so as to develop a Job Shop scheduling model based on the hierarchical optimization theory.

5.5.1.1 Principles of Hierarchical Optimization

Hierarchical optimization theory[48] is used to describe decision-making problems with hierarchical characteristics. The two-level optimization problem is the most basic hierarchical optimization problem, which is also known as the bi-level programming problem. In the case of the two-level optimization problem, first, an optimization parameter is determined by the upper decision-makers, and then with this optimal parameter, the objective is optimized by the lower decision-makers according to their own preferences. Second, based on the best response of the lower level, the overall optimal decision is made by the upper decision-makers as far as possible.

In addition, bi-level hierarchical optimization theory is introduced. Firstly, the upper level contains a central decision-maker, whilst the lower level consists of n parallel sub-systems. Secondly, the constraint information provided by the upper level is used by the lower programming problem, so as to feed back the optimal value obtained by the lower decision-makers to the upper level. Finally, the final decision in the global interest is determined by the upper level according to the optimal value obtained by the level.

The model for the bi-level hierarchical optimization theory is expressed as follows:

The upper level:

$$\max_{x \in X} F\left(x, v(x)\right) \tag{5-1}$$

where

$$X = \left\{x \in R^n \middle| h(x) \le 0\right\} \tag{5-2}$$

$$v(x) = \left(v_1(x), v_2(x), \cdots, v_n(x)\right) \tag{5-3}$$

The lower level:

$$v_i(x) = \max_{y^i \in Y_i(x)} f_i(x, y^i) i \in (1,2,\ldots,n) \tag{5-4}$$

where

$$Y_i(x) = \{y^i \in R \mid g^i(x, y^i) \le 0\} \tag{5-5}$$

Formula (5-1) represents the upper objective function, while formula (5-4) represents the lower objective function. Formula (5-2) and (5-5) are the upper and lower bound functions, respectively. x, y^i are decision variables, and $v(x)$ in formula (5-3) is a decision vector.

5.5.1.2 Job Shop Scheduling Model Based on Hierarchical Optimization Theory

The Job Shop scheduling model is developed based on the hierarchical optimization theory.[49–51] In order to facilitate the presentation, the following notation is used in the development of the mathematical model:

Parameters:

M = Machine set $\{1,2,\ldots m \ldots, M\}$,
S = Part set $\{1,2,\ldots s \ldots, S\}$;
N = Part task set $\{1,2,\ldots n \ldots, N\}$;
O_n^s = The number of parts required by part task n;
$T^{(m)}$ = Available working hours of machine m in the planning horizon;
t_s^m = the unit operation processing time for processing part s by using machine m;
C_s^m = the unit cost for processing part s by using machine m;
P_s^m = the production volume of part s in machine m;
D_n = the delivery due date of part n;
δ_T^n = the tardiness of part task n, which depends on each part task;
λ = the earliness of part task n;
T_n^m = the completion time for processing task n by using machine m;

Decision variables: $\sigma = \{\sigma_1, \ldots\ldots, \sigma_m\}$ = The set of allocation n parts to m machines, in which σ_m means the operation sequence of tasks assigned to machine m.

The tasks in Job Shop manufacturing systems are allocated in a globally optimal way. At the same time, there is need to maximize the interests of the resource capacity management Agent. Hence, in the procedure of the task allocation, the task assignment is globally optimized while the interests of the resource capacity management Agent are locally optimized. The positive feedback production scheduling method based on the hierarchical optimization theory is presented in detail in this section.

In the production management process, with the aid of the collaborative scheduling Agent, it is necessary to ensure that the task management Agent has the lowest possible scheduling costs. According to just-in-time (JIT) theory, scheduling costs include manufacturing costs, temporary inventory costs and tardiness costs. If the tasks are completed ahead of production schedules, this leads to temporary inventory costs. On the other hand, if the tasks are not completed before the delivery due dates, this results in tardiness costs. The tasks should be properly allocated to each machine so as to minimize the completion time of tasks incurred in each machine.

Indeed, the hierarchical optimization model governing the positive feedback production scheduling problem in Job Shop manufacturing systems has the following form:

<u>The upper level</u>: to minimize scheduling costs

$$\min \; Z = \sum_{m=1}^{M}\sum_{s=1}^{S} C_s^m P_s^m + \sum_{n=1}^{N} \delta_T^n \left(T_n^m - D_n\right) + \sum_{n=1}^{N} \lambda \left(D_n - T_n^m\right) \qquad (5\text{-}6)$$

where

$$T_n^m > D_{n,}\ \lambda = 0;\ \text{when}\ T_n^m < D_{n,}\ \delta_T^n = 0$$

<u>The lower level</u>: to minimize the completion time incurred in each machine

$$\min\left\{\max\left(T_n^m\right)\right\} = \sum_{n\in\sigma_m}\sum_{s=1}^{S} t_s^m O_n^s \quad \left(n \in N \quad m \in M\right) \qquad (5\text{-}7)$$

where

$$\sum_{s=1}^{S} t_s^m P_s^m \le T^{(m)} \qquad (5\text{-}8)$$

$$T_i^m - O_i^s t_s^m \geq T_j^m M T_i^m - O_i^s t_s^m \leq T_j^m \quad i,j \in N \tag{5-9}$$

$$P_s^m = \sum_{n \in \sigma_m} O_n^m \geq 0 \tag{5-10}$$

Formula (5-6) is the upper decision objective function of the collaborative scheduling Agent, which is to minimize scheduling costs for Job Shop manufacturing systems. Formula (5-7) is the lower optimization objective function of the collaborative scheduling Agent, which is to minimize the completion time incurred in the entire manufacturing system. Formula (5-8) is a constraint related to machine capacity. Formula (5-9) is a constraint that makes sure that the operation processing time to manufacturing a part is continuous. Formula (5-10) is a non-negative constraint.

5.5.2 Multi-Agent Positive Feedback Scheduling System Based on Contract Net Protocol

The positive feedback production scheduling process for Job Shop manufacturing systems is completed by using the collaborative scheduling Agent, the task management Agent and the resource capacity management Agent, simultaneously. The collaborative procedure of each Agent is illustrated in Figure 5-8. The tender method is adopted by the task management Agent to select machines to implement production tasks. The starting time and the completion time for processing each task are taken into consideration. In the implementation procedure of scheduling tasks, the scheduling instructions are adjusted by the task management Agent in accordance with changes in instructions of the upper level.

The procedure of production task allocation consists of three stages: 1) at the first stage, with the aid of the task management Agent, the bidding documents generated by tasks are allocated to the collaborative scheduling Agent. 2) at the second stage, the collaborative scheduling Agent is used to bid according to the capacity information provided by the resource capacity management Agent in the factory. 3) at the third stage, in accordance with the content and time requirements of services, the collaborative scheduling Agent is adopted to audit the

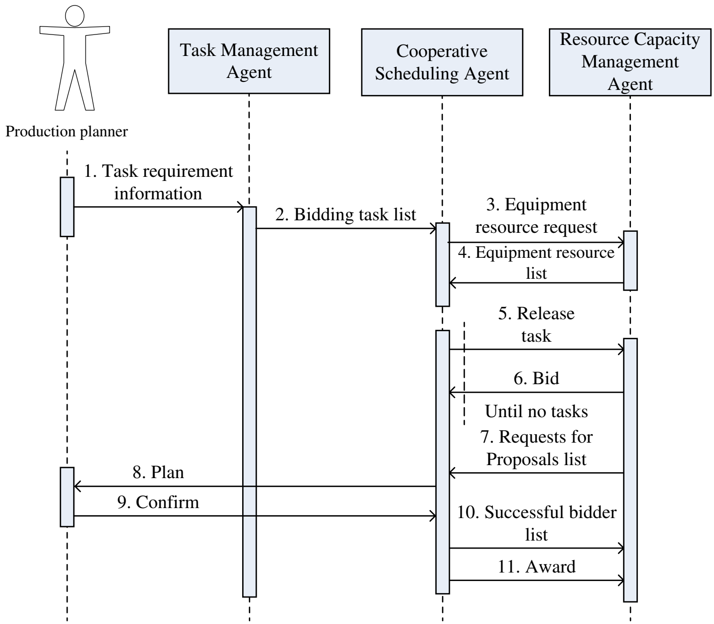

Figure 5-8 Multi-Agent Positive Feedback Scheduling System based on Contract Net Protocol.

bidding documents developed by the resource capacity management Agent. The successful bidder is determined, and the tenders are awarded in terms of the contract net protocol. The collaborative process of the positive feedback scheduling driven by tasks is generated by the above three stages.

5.5.3 Positive Feedback Production Scheduling Algorithm Based on the Hierarchical Genetic Algorithm

With the aid of its internal knowledge base, the collaborative scheduling Agent is used to develop effective ways of optimally solving production scheduling problems for Job Shop manufacturing systems. In particular, the techniques of hierarchical genetic algorithms are adopted in the positive feedback production scheduling optimization method. As a logic inference knowledge or method, this approach is packaged in the knowledge base of the collaborative scheduling Agent.

Genetic algorithms are stochastic search algorithms based on the mechanism of probability and global optimization. Genetic algorithms provide many advantages over other methods: to name a few, good flexibility, global search ability, universality, scalability, implicit parallelism, robustness, a good combination capacity with computers, and impressive resulting search performance in solving large-scale problems. In this section, a genetic algorithm is used as an optimization tool to solve hierarchical positive feedback scheduling problems. The task management Agent is adopted to allocate tasks, while the resource management Agent is applied to receive tasks. When the production schedules are decided, based on hierarchical optimization theory, the upper task management Agent is used by obtaining machine capacity data to develop the feasible solution set σ. The lower task management Agent is adopted to solve the function $\min\{T_n^m\}$ with parameter σ so as to minimize the completion time of each machine, and to feed the results back to the upper task management Agent. According to the results provided by the lower task management Agent, the upper task management Agent is employed to adjust σ until the optimal solution Z is found. The techniques of genetic algorithms are adopted by the upper task management Agent in combination with the lower task management Agent to develop the feasible solution set and to optimize the objective function.

The hierarchical based genetic algorithm comprises the following elements:

5.5.3.1 Structure of the Individuals

The real-number matrix encoding method is used to represent feasible solutions to the positive feedback production scheduling problem. Each chromosome σ is represented with a real two-dimensional matrix $[\ ]_{m\times n}$, whose size is $m*n$. For instance, assume that there are three machines (m = 3) denoted as M1, M2, M3 and five tasks (n = 5) in a Job Shop manufacturing system. This set of tasks is encoded by using the matrix encoding method. As shown in the following matrix, each row in the matrix represents machine M1, M2, M3, respectively. Each column in the matrix represents the task from 1 to 5, sequentially. The second row in the matrix below indicates that the sequence of operations in machine M2 is task 4, task 1, task 3, task 5, task

2. If there is no need to process the task on the machine, then the operation processing time required in the machine is zero.

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 1 & 3 & 5 & 2 \\ 3 & 2 & 1 & 4 & 5 \end{bmatrix}$$

5.5.3.2 Initialization

Although the initial population of individuals is generated randomly in traditional genetic algorithms, in this section, the individuals that will appear in the initial population have to be determined by using the upper level. First, a group of individuals is randomly generated. Second, according to the constraints (5-2) related to the upper level, the selection operator then selects individuals from the randomly generated individuals, and places them into the mating pool. The selected individuals must satisfy the constraints (5-2). Assume that the number of chromosomes is R, and then the initial population consists of R individuals.

5.5.3.3 The Lower Level Feedback

This process is a unique one for the genetic algorithm used in the bi-level programming optimization problem. In the initialization process, R chromosomes are generated for the upper problem. The lower problem $\min\{T_n^m\}$ is minimized with the parameter σ so as to obtain the minimum value of $\min\{T_n^m\}$, and the obtained value of $\min\{T_n^m\}$ is fed back to the upper objective function Z.

5.5.3.4 Fitness Function

The fitness function is used to evaluate the merits of the individual in the genetic algorithm evolution process and is also the survival standard (e.g., the fittest individuals dominate over weaker ones). In this section, the fitness function is the upper objective function presented in section 5.5.1. The chromosomes are sorted from good to bad by their fitness function values. The smaller the value of Z, the better the optimal chromosome and the smaller the number.

5.5.3.5 Selection Operation

According to the standard "the fittest individuals dominate over weaker ones", the best individuals with higher fitness function values will be selected out from the current population, in order to increase the number of the best individuals in the population. Hence, the evolution process will move in a more optimal direction. Although there are many different selection schemes to implement genetic algorithms, the regular geometric selection scheme is used. This scheme is given by the following equation:

$$p = q / \left(1 - (1 - q)^{r}\right) \tag{5-11}$$

In formula (5-11), q is the probability of selecting the best individuals, and r is the serial number of the fitness function value. The smaller the serial number of the fitness function value, the better the individual.

5.5.3.6 Crossover Operation

In terms of the crossover rate pc, the individuals are selected and then placed into a mating pool to perform the crossover operation. In the crossover operation, the individual with the highest fitness value should be reserved and placed into the mating pool in order to avoid detroying the excellent sequence of genes. Hence, the excellent sequence of genes should be picked out first. For instance, if the parent chromosomes are encoded by using the following matrix representations,

$$\sigma_1 - \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 1 & 3 & 5 & 2 \\ 3 & 2 & 1 & 4 & 5 \end{bmatrix}$$

$$\sigma_2 - \begin{bmatrix} 1 & 3 & 2 & 4 & 5 \\ 4 & 3 & 1 & 2 & 5 \\ 3 & 1 & 2 & 5 & 4 \end{bmatrix}$$

after decoding these two chromosomes, it is known that there is no tardiness cost for task 1 and task 2 in the parent chromosome σ_1, and there is no tardiness cost for task 1 and task 3 in the parent chromosome σ_2. Hence, the excellent sequence of genes (1, 2, 3) is selected and reserved. Then, the offspring chromosomes are generated by the crossover operation. The offspring chromosomes thus become:

$$\sigma_1' - \begin{bmatrix} 1 & 3 & 3 & 4 & 5 \\ 4 & 3 & 3 & 5 & 5 \\ 3 & 1 & 2 & 5 & 5 \end{bmatrix}$$

$$\sigma_2' - \begin{bmatrix} 1 & 3 & 3 & 4 & 5 \\ 4 & 3 & 3 & 2 & 5 \\ 3 & 2 & 1 & 5 & 5 \end{bmatrix}$$

Each row of genes in the offspring chromosome is checked to verify the correctness of chromosomes and to ensure that each task in a row occurs only once. After the crossover operation is executed, non-crossover genes perform adaptive changes if the crossover genes conflict with non-crossover genes. Then new offspring chromosomes are generated after performing this checking operation.

$$\sigma_1' - \begin{bmatrix} 1 & 3 & 2 & 4 & 5 \\ 4 & 3 & 1 & 2 & 5 \\ 3 & 1 & 2 & 5 & 4 \end{bmatrix}$$

$$\sigma_2' - \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 1 & 3 & 2 & 5 \\ 3 & 2 & 1 & 4 & 5 \end{bmatrix}$$

5.5.3.7 Mutation Operation

When the crossover process is completed, the mutation operator will be used to improve the probability for generating the best individuals so as to improve the quality of the entire population. It is easy to implement the mutation operator based on the real-number encoding method. The simple way is to exchange two different genes or multiple genes in the parent chromosome. For instance, if a parent individual is encoded by using the following matrix

$$\begin{bmatrix} 1 & 3 & 2 & 4 & 5 \\ 4 & 3 & 1 & 2 & 5 \\ 3 & 1 & 2 & 5 & 4 \end{bmatrix}$$

The mutation operator is employed to reverse a randomly selected row in the parent chromosome. If the third row is selected, whenever the mutation operations are triggered, the resulting individual will become

$$\begin{bmatrix} 1 & 3 & 2 & 4 & 5 \\ 4 & 3 & 1 & 2 & 5 \\ 4 & 5 & 2 & 1 & 3 \end{bmatrix}$$

The basic procedure of the hierarchical based genetic algorithm is presented below:

Step 1: Set the initial values of parameters, such as population size *R*, crossover rate *pc*, mutation rate *mu*, and maximum number of iterations *Gen*.

Step 2: Generate initial population of *R* individuals randomly.

Step 3: For each individual, the linear programming subroutine is used to solve the lower-level optimization problem, and the results are then input into the fitness function to calculate the corresponding fitness value.

Step 4: Sort the individuals by their fitness function values. Place the best individual with the highest fitness function value in the upper level in the first position. Select individuals from the population *R* times.

Step 5: Pair up the individuals in the mating pool and generate new-born offspring individuals by using the operators of crossover and mutation.

Step 6: Check the pre-specified stopping condition. If it is satisfied, terminate the search process, and return the overall best solution as the final solution. Otherwise, go to *Step 3*.

Step 7: Submit the final solution to decision-makers. If it is satisfied, then output the final solution. Otherwise provide compromise objective function values and its corresponding satisfaction degree or feasibility, and go to *step 4*.

5.5.4 Case Study

In this section, the hierarchical optimization theory–based genetic algorithm is proposed for solving the positive feedback scheduling optimization problem for Job Shop manufacturing systems. An Intel PIII–based computer is used to run the software package developed by using Visual Studio 2008. The positive feedback scheduling optimization method is encapsulated in the collaborative scheduling Agent as logic inference knowledge of the Agent controller. And its information interaction function with other Agents is implemented by the object function.

The practical data collected from a company manufacturing compressor are used to demonstrate the effectiveness of the proposed hierarchical optimization theory–based genetic algorithm. The manufacturing system contains five production lines identified by using a numerical number, ranging from 1 to 5. These five production lines are used to produce the four different types of compressors, namely, $S/Q/V/D$. In this study, the unit scheduling time in this manufacturing system is a day. Table 5-1 shows the list of incoming tasks. The unit operation processing time is summarized in Table 5-2. The unit cost information for four products is shown in Table 5-3. Table 5-4 shows the available hours in each machine.

The genetic parameters to be adopted for the hierarchical optimization–based genetic algorithm are as follows: Maximum number of iterations = 1000, Population size =50, Crossover rate = 0.8, Mutation rate =0.15. Table 5-5 presents the final positive feedback production scheduling results by using the hierarchical optimization–based genetic algorithm. The results show that the proposed scheduling approach is not only effective in providing a good balance between costs and machine utilization for the relationship between shop floor management and shop floor, but is also very efficient in achieving automation and optimization for production scheduling in the assembly shop floor.

Table 5-1 A list of incoming tasks.

Task No. (N)	A101	A102	A103	A104	A105	A106	A107	A108	A109	A110	A111
Type (S)	S	Q	V	Q	S	D	Q	V	Q	S	D
Production volume (o_n^s)	2000	1000	4000	3500	4500	3500	2000	1000	5000	5000	4500
Delivery due date (D_n/day)	4	3	6	5	7	2	4	5	6	3	2
Tardiness penalties coefficient (δ_T^n)	5.4	4.7	6.2	3.7	1.2	4.2	1.2	4.5	1.5	3.5	6.7

Task type	Machine 01	Machine 02	Machine 03	Machine 04	Machine 05
S	22	25	20	20	21
Q	20	19	20	21	22
V	25	28	20	20	21
D	18	17	19	24	22

Table 5-3 Unit cost of products C_s^m ($).

Product type	Machine 01	Machine 02	Machine 03	Machine 04	Machine 05
S	22.5	25	20	20	21
Q	20	19	20	21	22
V	25	28	20	20	21
D	18	17	19	24	22

Table 5-4 The available hours in each machine C_s^m (daily working hours, h).

	Machine 01	Machine 02	Machine 03	Machine 04	Machine 05
Available hours	10	8	8	10	10

Table 5-5 The final positive feedback production scheduling results.

Task No.	Delivery due date	Arrange time	Cost	Tardiness cost
A101	144000	60000	40000	0
A102	108000	93000	20000	0
A103	216000	172000	88000	0
A104	180000	137500	71000	0
A105	252000	115500	93500	0
A106	72000	42000	73000	0
A107	144000	121000	40000	0
A108	180000	135000	20000	0
A109	216000	144000	98000	0
A110	108000	100000	102000	0

5.6 Negative Feedback–Based Production Rescheduling in Job Shop Manufacturing Systems

When dynamic events occur in the environment of Job Shop manufacturing systems, the collaborative scheduling Agent is adopted by analyzing the on-line information provided by Multi-Agent production control system to organize the task management Agent and the resource capacity management Agent to carry out negative feedback rescheduling and to adjust to the production plan. In this section, the information-processing requirements in negative feedback rescheduling are performed by using the collaborative scheduling Agent. The ant colony auction protocol based negative feedback rescheduling algorithm is proposed.

5.6.1 Problem Description

In the positive feedback production scheduling process for Job Shop manufacturing systems, the collaborative scheduling Agent is used to allocate production tasks to different machines in order to optimize the performance of the manufacturing system. Tasks are assigned based on the determinant task set. Suppose that capacity parameters, equipment maintenance parameters and machining time parameters in the manufacturing systems are known. However, there are a lot of dynamic events in the actual manufacturing environment, for instance, a shortage of raw materials, new tasks, changes of delivery due dates, personnel adjustments inside the shop floor, machine breakdown and so on. In general, these dynamic events can be briefly classified into four categories:[52] 1) task-related events; 2) resource-related events; 3) process-related events; 4) others. The entire manufacturing process becomes a dynamic process while dynamic events occur. It is necessary to introduce the appropriate way to solve this kind of problem. In this book, due to space limitations, we use only the first type of dynamic events with inserting unplanned task to illustrate it.

In order to respond rapidly to various dynamic events in manufacturing systems and to reduce their impact on determinant production plans, response scheduling methods and periodic rescheduling methods have been proposed. Response scheduling highlights the ability to respond when the environment changes, which is an approach or a response mechanism for rescheduling. Periodic rescheduling is an intermittent scheduling method with a period of time as a scheduling period. The complexity of these dynamic scheduling problems has led to the recent interest in theory and engineering in investigating on how to deal with the production scheduling problem in a dynamic production environment with dynamic events; there are various ways to solve this problem. In this book, a reactive negative feedback rescheduling method is developed.

In the negative feedback rescheduling process for Job Shop manufacturing systems, when extra tasks arrive at the manufacturing system, the blackboard system is used to develop topological relationships among resources, shown in Figure 5-9, which helps the task management Agent to choose proper resources according to resource capacity in the manufacturing system.

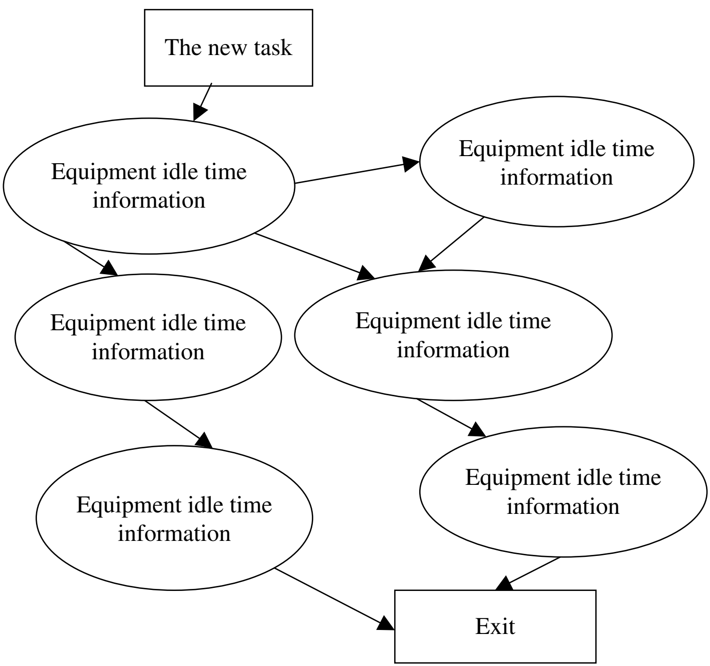

Figure 5-9 The Topology model of resources in the shop floor.

The collaborative scheduling Agent is adopted to choose a reasonable path in the topological model to ensure the minimal change to the original task, and to ensure the shortest completion time after the new task has entered the manufacturing system. Rather than parameters adopted for the positive feedback scheduling problem, some additional ones that are adopted for the negative feedback rescheduling problem are presented as follows:

$$EN\,\mathrm{k} = \text{the task set of unplanned parts} \left\{e1, e2, \ldots en \ldots, eN\right\}.$$

The negative feedback rescheduling problem in Job Shop manufacturing systems is expressed by using a graphical model

$$G = (V, E)$$

Where *V*: Point set of the available time of resource;

E: Edge set of the edges connecting any two points in *V*;

The aim of the negative feedback rescheduling problem in Job Shop manufacturing systems is to look for a combination of a group of edges in *V* to minimize the changes in the original tasks, and to ensure the shortest completion time after the new task has entered the manufacturing system.

5.6.2 Multi-Agent Negative Feedback Rescheduling System Based on Ant Colony Auction Protocol

In the real world, a large number of ants take approximately a straight path between their nest and the food location (food sources) rather than curves or other shapes, as shown in Figure 5-10(a). If there is an obstacle on the moving track of the ant colony, shown in Figure 5-10(b), or if there is an obstacle between the nest and the food source – a more likely situation – ants encountering the obstacle at the beginning will move up and down uniformly to bypass the obstacle, that is to say, regardless of the length of the path, the ants select a feasible path with the same probability. In the travelling process, ants deposit a pheromone on their trajectory and determine the amount of pheromone to guide their direction and judge on a variety of options. Ants tend to choose a path marked with high pheromone

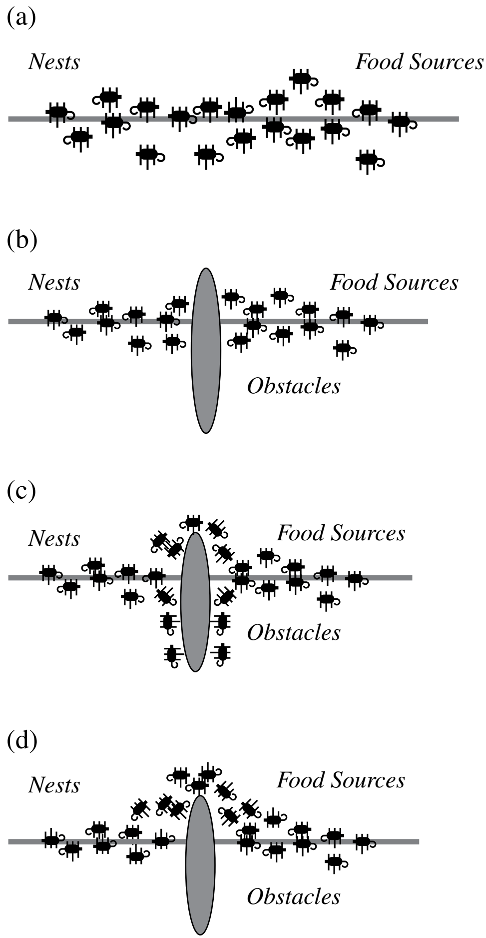

Figure 5-10 The foraging behaviour of real ants.

concentrations. In the case of the same time length and the equal probability (the same number of ants), ants spend less time to go back on the shorter track so that more pheromone is left on the shorter path, and gradually more and more ants choose the shorter path, as shown in Figure 5-10(c). Finally, it is obvious that almost all the ants move on the shortest path, shown in Figure 5-10(d). The pheromone plays a key role in the activity of ants.

The ant colony auction protocol is a process to execute negotiations by simulating the information-exchange process of an ant colony's foraging behaviour. In this section, in view of the foraging behaviour of real ants, the ant colony auction protocol–based negative feedback rescheduling algorithm is investigated for Job Shop manufacturing systems. The negotiation activities[53–55] among Agents in a Multi-Agent system are designed to guide information interaction among the collaborative scheduling Agent, the task management Agent, the resource capacity management Agent, and the production control Agent in the scheduling layer. The negotiation process among Agents is illustrated in Figure 5-11.

The aim of negotiation-based MAS using the ant colony auction protocol is to allocate each unplanned task to its corresponding machine. First, the information and parameters in dynamic events are obtained by the production control Agent. Second, feasible paths are provided to the collaborative scheduling Agent by multiple resource capacity management Agents that can complete the same tasks. Third, pheromone concentrations, other parameters on paths and the transition probability are evaluated by using the collaborative scheduling Agent. Finally, the results are released to the dispatcher, and the resource capacity management Agent is awarded after the dispatcher confirms it.

5.6.3 Ant Colony Algorithm–Based Negative Feedback Rescheduling Approach

As a negative feedback rescheduling optimization method, the ant colony algorithm is encapsulated in the knowledge base of the collaborative scheduling Agent.

The ant colony algorithm is inspired by the foraging behaviour of real ants in nature, and the artificial ant is defined as an abstraction of real ants. Owing to simulating the foraging behaviour of ants in nature, the ant colony algorithm is an application mechanism. Therefore, it is necessary to abstract real ants, and it is neither possible nor necessary to fully reproduce real ants. Its purpose is to more effectively portray the mechanism of the real ant colony that could be used by algorithms and to abandon the factors unrelated to the algorithm model. According to the

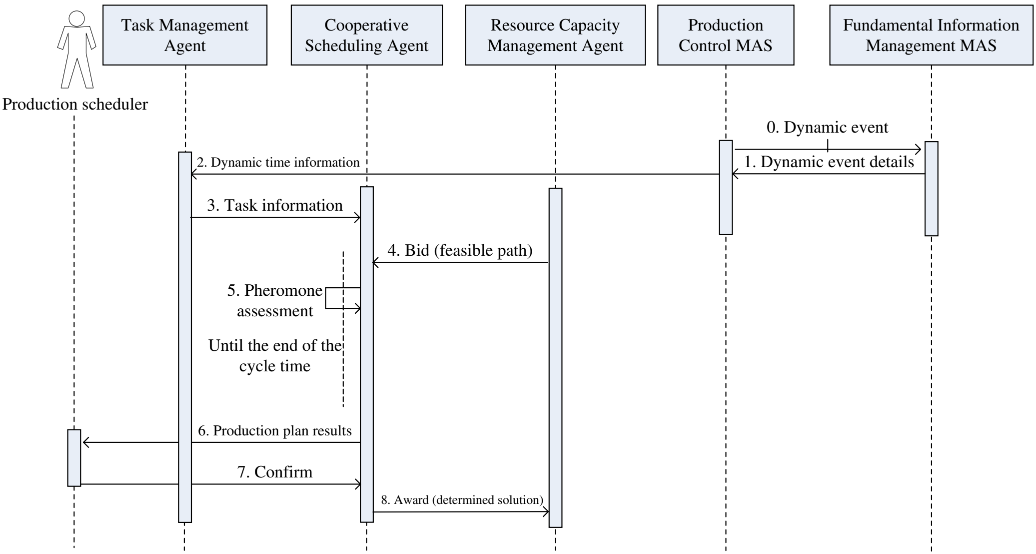

Figure 5-11 The negotiation model among Agents in a Multi-Agent negative feedback rescheduling system for Job Shop manufacturing systems.

principle of bionics, the prototype of the ant colony algorithm consists of the following three major components:

1) When a path is explored, the amount of pheromone is deposited on the path;
2) When an ant is at a starting node, the next node is selected according to its respective selection probability;
3) Set the tabu list: ants are not allowed to select passed nodes or nodes unsatisfied with executable conditions.

Suppose that $b_i(t)$ denotes the quantity of ants at node i at the moment t, and $\tau_i(t)$ denotes the amount of pheromone on the path (i, j). n represents the scale of problem, m is the total number of ant colony. Then,

$$m=\sum_{i=1}^{n} b_i(t) \tag{5-12}$$

$$T=\{\tau_{ij}(t) \mid c_i, c_j \subset C\} \tag{5-13}$$

T is a set of the amount of remaining pheromone on the path l_{ij} in set C at the moment t. At the initial moment, the amount of pheromone on all paths is equal and is expressed by the following equation:

$$\tau_{ij}(0)=const \tag{5-14}$$

In the movement process of Ant $k(1,2,\ldots K)$, its moving direction is determined by the amount of pheromone on each path. The tabu list $\Gamma^k(1,2,\ldots,K)$ is used to record the visited points. Γ^k is dynamically adjusted as the ants move on. In the search process, the state transition probability of ants is calculated in accordance with the amount of pheromone and heuristic information on each path. At the moment t, the state transition probability that ant k chooses to mover from node i to node j is:

$$p_{ij}^k(t)=\begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}(t)]^{\beta}}{\sum\limits_{s \notin \Gamma^k} [\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}(t)]^{\beta}} & if\ j \notin \Gamma^k \\ 0 & if\ j \in \Gamma^k \end{cases} \tag{5-15}$$

where α is the information heuristic factor that indicates the relative importance of pheromone accumulated in the movement process and changes the dependence of ants on the existing pheromone. The larger the α value, the greater the impact of path information on the decision-making of ants, the more the probability selection is consistent with collaboration among ant colony. β is the expectation heuristic factor that represents the relative importance of the heuristic information on the path selection and the dependence on visibility (such as path distance). The larger the β value, the greater the impact of the path distance on ant selection, the more the probability selection consists with the greedy rule.

$n_{ij}(t)$ is the value of the heuristic function, which is calculated by using the following formula:

$$n_{ij}(t)=\frac{1}{d_{ij}} \quad (5\text{-}16)$$

In formula (5-16), d_{ij} denotes the distance l_{ij} between two adjacent nodes. For ant k, the smaller d_{ij} is, the larger $n_{ij}(t)$ is. Excessive residual pheromone trail on certain paths directly dispatches the probability selection so as to cause undesirable effects. In order to avoid excessive residual pheromone, the residual pheromone is globally updated after each ant has visited all the nodes (n nodes). At the moment $t+n$, the pheromone on the path (i,j) is updated according to formula (5-17):

$$\tau_{ij}(t+n)=(1-\rho)\cdot\tau_{ij}(t)+\Delta\tau_{ij}(t) \quad (5\text{-}17)$$

$$\Delta\tau_{ij}(t)=\sum_{k=1}^{m}\Delta\tau_{ij}^{k}(t) \quad (5\text{-}18)$$

Where ρ is the pheromone evaporating rate and $1-\rho$ is the residual pheromone factor. In order to prevent unlimited accumulation of pheromone, ρ is to be $\rho\subset[0,1]$. $\Delta\tau_{ij}(t)$ is the pheromone increment on the path (i,j) in this cycle, and $\Delta\tau_{ij}=(0)=0$ at the initial time. $\Delta\tau_{ij}^{k}(t)$ is the pheromone increment of ant k on the path (i,j) in this cycle.

According to the pheromone update strategy, Dorigo proposed three different basic ant colony models,[56] respectively called

Ant-Cycle model, Ant-Quantity Model and Ant-Density model. Their difference lies in how to achieve $\Delta\tau_{ij}^{k}(t)$.

a) In Ant-Cycle model:

$$\Delta\tau_{ij}^{k}(t)=\begin{cases}\dfrac{Q}{d_{ij}} & \textit{if ant k traverses}\,(i,j)\,\textit{in this cycle}\\ 0 & \textit{other}\end{cases} \tag{5-19}$$

Where Q is the pheromone intensity that affects the convergence speed of the ant colony algorithm. L_K is the total length of the path visited by ant *k* in this cycle.

b) In Ant-Quantity Model:

$$\Delta\tau_{ij}^{k}(t)=\begin{cases}\dfrac{Q}{d_{ij}} & \textit{if ant k traverses}\,(i,j)\,\textit{between time t and time}\,t+1\\ 0 & \textit{other}\end{cases} \tag{5-20}$$

c) In Ant-Density Model:

$$\Delta\tau_{ij}^{k}(t)=\begin{cases}Q & \textit{if ant k traverses}\,(i,j)\,\textit{between time t and time}\,t+1\\ 0 & \textit{other}\end{cases} \tag{5-21}$$

In the Ant-Quantity model and the Ant-Density model, the local information is used by the pheromone: that is to say, whenever an ant completes a path, the pheromone trail on the path it has visited will be updated. In the Ant-Cycle model, the global information is used by the pheromone: that is to say, after all ants have completed all paths (a cycle), the pheromone trails on all paths they have visited will be updated. Among these three models, it is noted that Ant-Cycle model achieves good results in application. In general, it is regarded as the basic model in the ant colony algorithm.

According to the characteristics of the Agent, which include autonomy, communicativeness, reactivity, adaptability and mobility, and so on,[57] each Agent in Job Shop manufacturing systems is adopted to simulate the ant colony foraging process to optimize scheduling objectives affected by dynamic events.

The procedure of dynamic scheduling based on the ant colony algorithm is presented below:

Step 1. Tasks arrive. The task management Agent is used to receive them and assign a task ant for each part task. The ants bred by the task management Agent are employed to record corresponding part tasks information formatted by using a one-dimensional array *procedureName[stepNum],* where *stepNum* is the process number and *procedure-Name* is the name of each process.

Step 2. The resource capacity management Agent is adopted to breed ants; the number of ants is equivalent to the number of machines. Then the resource capacity management Agent is used to record the machining capacity information of machines, respectively, and to return resource information presented by using the following format *(machineNum, productName, stepNum).* That is to say, the process information of a part task could be processed by using a machine. And this information is transmitted to the collaborative scheduling Agent by the resource capacity management Agent.

Step 3. The task management Agent is used to draw the topology graph of the Job Shop manufacturing system in accordance with the resource information, and feasible solutions are explored by task ants in the topology graph. The output information of each task is expressed by using a two-dimensional array *productName [stepNum] [machineNum]*, which denotes the process of a task is completed by using an assigned machine. The task Ant is at any node in the shop floor topology; a machine is selected according to its respective transition probability as shown in equation (5-22). The probability for selecting a machine from the set of available machines is

$$p_{jk} = \frac{\tau_{jk}}{\sum_{k \in \mathrm{K}} \tau_{jk}} \quad M \tag{5-22}$$

Where K is the set of all available machines, and τ_{jk} is the amount of pheromone trail on the path from machine j to machine k.

Step 4. The collaborative scheduling Agent is used to arrange the time value sequence obtained by ants in Gantt chart format and to calculate the minimal completion time in the Gantt chart outputted by the task management Agent. Then the collaborative scheduling Agent is adopted to evaluate the information and to dispatch ants to deposit the proper amount of pheromone. The ants corresponding to $Min[t(o_{End})]$ deposit a greater amount of pheromone; while the ants corresponding to larger $t(O_{END})$ deposit smaller amount of pheromone. In this book, the Ant-Cycle model[58] is used, and the amount of pheromone is defined by applying the following equation

$$\Delta\tau = \frac{Q}{L_i} \tag{5-23}$$

Where Q is a constant and L_i denotes the completion time in one schedule outputted by the monitoring Agent *i*.

Step 5. Due to the feature that the pheromone trail on the paths ants have visited will be reduced in the process of ant colony foraging, α denotes the pheromone evaporating rate. When ants haven't visited a path for a long time, the amount of pheromone on this path will be reduced to zero. After an ant has completed a tour, the update of the pheromone trail on the path is performed by applying the rule

$$\tau_{jk}^{i}(t+\Delta t) = \alpha\tau_{jk}(t) + \Delta\tau \tag{5-24}$$

Step 6. Return to *Step 1*.

In the procedure of manufacturing the current task, if a new task arrives, the ant colony auction mechanism will perform the follow operations:

1) Add a new part task *i*, an ant is assigned for each task, and the assigned ants are used to record the process information.
2) The new task ant and the current task ant simultaneously start to look for feasible paths according to the above procedure.

Table 5-6 A list of new incoming tasks.

Task No. (N)	A112	A113
Type (S)	S	Q
Production volume (o_n^s)	2000	2000
Delivery due date (D_n)	4	3
Tardiness penalty coefficient (δ_T^n)	6.4	4.1

5.6.4 Case Study

In this section, a Multi-Agent system based on the ant colony auction protocol is developed, and an ant colony algorithm based negative feedback rescheduling approach is proposed. The practical data presented in section 5.5.4 are used to demonstrate the effectiveness of the proposed Multi-Agent negative feedback rescheduling system based on the ant colony auction protocol. Suppose that the production process in a compressor manufacturing system is executed according to the results presented in section 5.5.4. New orders are inserted after 20 hours, shown in Table 5-6.

The ant colony auction protocol–based negative feedback rescheduling approach is applied to generate an optimal production schedule. Table 5-7 summarizes the results of the ant colony auction protocol–based negative feedback rescheduling approach.

Since new task has been inserted, some previous planned tasks should be adjusted. Some of the adjusted tasks could not satisfy their delivery due dates, which leads to tardiness costs. Table 5-7 indicates that new tardiness costs caused by inserting new tasks are 39,350 units. The optimal production schedule is determined by manufacturing system administrators according to calculated planning results and cost data.

5.7 Conclusion

This chapter has presented the application of Agent technology for production scheduling in Job Shop manufacturing systems. Firstly, considering the requirements of the dynamic complex

Table 5-7 Results of the ant colony auction protocol based negative feedback rescheduling approach.

Order No.	Delivery Due date	Original arrange time	Present arrange time	Original cost	Present cost	Original tardiness cost	Present tardiness cost
A101	144,000	60,000	60,000	40,000	40,000	0	0
A102	108,000	93,000	163,000	20,000	20,000	0	4700
A103	216,000	172,000	210,000	88,000	80,000	0	0
A104	180,000	137,500	144,000	71,000	76,000	0	0
A105	252,000	115,500	188,000	93,500	102,000	0	0
A106	72,000	42,000	42,000	73,000	73,000	0	0
A107	144,000	121,000	163,000	40,000	40,000	0	4500
A108	180,000	135,000	135,000	20,000	20,000	0	0
A109	216,000	144,000	143,000	98,000	98,000	0	0
A110	108,000	100,000	177,000	102,000	98,000	0	30,150
A111	72,000	68,000	68,000	77,000	77,000	0	0
A112	144,000		140,000		40,000		0
A113	108,000		106,000		39,000		0

manufacturing environment, a Multi-Agent double feedback–based scheduling system is developed in this chapter. Next, the positive feedback production scheduling scheme for Job Shop manufacturing systems is proposed based on the hierarchical optimization theory. The ant colony auction protocol–based negative feedback rescheduling algorithm is designed. Finally, the collaborative process of each Agent and the information processing methods of the collaborative scheduling Agent for production scheduling in Job Shop manufacturing systems are presented in detail. The practical examples are used to demonstrate the effectiveness of the double feedback scheduling strategy.

References

[1] Zhang Jie, Wang Ningsheng, DuanZhengcheng, and Li Peigen. The Principle of Automatic Generation of 0bject-Oriented FMS Dynamic Scheduling Software. *Journal of Huazhong University of Science & Technology*. 1997, 25(11): 16–19.

[2] Brussel, H. V. Holonic Manufacturing Systems, The Vision Matching the Problem. *Proceedings of the 1st European Conference on Holonic Manufacturing Systems*. Hannover, Germany. 1994.

[3] Galbraith, J. *Designing Complex Organizations*. Reading, MA: Addison-Wesley, 1973.

[4] Peres, D., and Paulli, J. An Integrated Approach for Modeling and Solving the General Multiprocessor Job-Shop Scheduling Problems Using Tabu Search. *Annals of Operations Research*, 1997, 70(2): 281–306.

[5] Srinoi, P., Shayan, E., and Ghotb, F. A Fuzzy Logic Modeling of Dynamic Scheduling in FMS. *International Journal of Production Research*, 2006, 24(11): 1–21.

[6] Yu, L., Shin, M. H., and Sekiguchi, T. Fuzzy Inference-Based Multiple Criteria FMS Scheduling. *International Journal of Production Research*, 1999, 37(10): 2315–2333.

[7] Wan Guohua, Using Fuzzy Scheduling System for Solving a Dynamic Job Shop Problem. *Systems Engineering Theory & Practice*, 2001(8): 97–101.

[8] Mastrolilli, M., and Gambardella, L. M. Effective Neighborhood Function for Job Shop Scheduling Problems. *Journal of Scheduling*, 2002, 3(1): 3–20.
[9] Hu Yongmei, Jia Lei, and Li Qiqiang, Real-time Scheduling Modeling Method Based on S-rough sets. *System Engineering and Electronics*, 2004, 26(10): 1418–1420.
[10] Li Li, Cheng Ye, and Yuan Shouhua, Research on Priority Scheduling Algorithm Based on Bottleneck Analysis. *Computer Integrated Manufacturing Systems*, 2005, 11(2): 247–250.
[11] Blackstone, J. H., Phillips, D. T., and Hogg, G. L. A State-of-the-Art Survey of Dispatching Rules for Manufacturing Job Shop Operations. *International Journal of Production Research*, 1982, 20: 27–45.
[12] Kiran, A. S., and Smith, M. S. Simulation Studies in Job Shop Scheduling: A Survey. *Computers and Industrial Engineering*, 1984, 8: 46–51.
[13] Montazeri, M. V., and Wassenhove, L. N. Analysis of Scheduling Rules for an FMS. *International Journal of Production Research*, 1990, 28: 785–802.
[14] Li Sufen, Zhu Yunlong, and Yin Chaowan. Flow Shop Scheduling with Stochastic Processing Times and Machine Breakdowns. *Computer Integrated Manufacturing Systems*, 2005, 10(11): 1425–1429.
[15] Li Shugang, Wu Zhiming, and Pang Xiaohong. Job Shop Scheduling in Real-Time Cases. *Journal of Shanghai Jiao Tong University*, 2004, 38(4): 559–563.
[16] Kim, M. H., and Kim, Y. D. Simulation-Based Real Time Scheduling Mechanism in a Flexible Manufacturing System. *Journal of Manufacturing Systems*, 1994, 13: 85–93.
[17] Shafaei. R., and Brunn, P. Workshop Scheduling Using Practical (Inaccurate) Data Part 2: An Investigation of the Robustness of Scheduling Rules in a Dynamic and Stochastic Environment. *International Journal of Production Research*, 1999, 37(18): 4105–4117.
[18] Fox, M. S., and Smith, S. F. ISIS: A Knowledge-Based System for Factory Scheduling. *Expert System*, 1984, 1(1): 25–49.
[19] Tang, L., Liu, W., and Liu, J. A Neural Network Model and Algorithm for the Hybrid Flow Shop Scheduling Problem in a Dynamic Environment. *Journal of Intelligent Manufacturing*, 2005, 16: 361–370.

[20] Michael, J. S. A Neural-Net Approach to Real-Time Flow-Shop Sequencing. *Computer & Industrial Engineering*, 2000, 38(2): 125–147.

[21] Masahiko, A. An Artificial Neural Network Optimized by a Genetic Algorithm for Real-Time Flow-Shop Scheduling. *Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies*. 2000: 329–332.

[22] Cowling I. P., Ouelhadj D., and Petrovic S. A Multi-Agent Architecture for Dynamic Scheduling of Steel Hot Rolling. *Journal of Intelligent Manufacturing*, 2003, 14: 457–470.

[23] Aydin, E. M., and Öztemel, E. Dynamic Job-Shop Scheduling Using Reinforcement Learning Agents. *Robotics and Autonomous Systems*, 2000, 33: 169–178.

[24] Qiao Bing, Sun Zhijun, and Zhu Jianying, Distributed Dynamic Job Shop Scheduling Based on Agent. *Information and Control*, 2001, 31(4): 292–296.

[25] Shnits, B., Rubinovitz, J., and Sinreich, D. Multi-Criteria Dynamic Scheduling Methodology for Controlling a Flexible Manufacturing System. *International Journal of Production Research*, 2004, 42(17): 3457–3472.

[26] Kouiss, K., and Pierreval, H. Using Multi-Agent Architecture in FMS for Dynamic Scheduling. *Journal of Intelligent Manufacturing*, 1997, 8: 41–47.

[27] Potts, C. N. Analysis of a Linear Programming Heuristic for Scheduling Unrelated Parallel Machines. *Discrete Applied Mathematics*, 1985, 10(2): 155–164

[28] Shim, S.O., and Kim, Y.D. A Branch and Bound Algorithm for an Identical Parallel Machine Scheduling Problem with a Job Splitting Property. *Computers & Operations Research*, 2008, 35(3): 863–875

[29] Jenabi, M. Two Hybrid Meta-Heuristics for the Finite Horizon ELSP in Flexible Flow Lines with Unrelated Parallel Machines. *Applied Mathematics and Computation*, 2007, 186(1): 230–245.

[30] Omar, M. K., and Teo, S.C. Minimizing the Sum of Earliness/Tardiness in Identical Parallel Machines Schedule with Incompatible Job Families: An Improved MIP Approach. *Applied Mathematics and Computation*, 2006, 181(2): 1008–1017.

[31] Gao Jiaquan and Fang Lei. Parallel Genetic Algorithm for Solving Non-Identical Parallel Multi-Machine Schedule Problem. *Computer Engineering*, 2007, 33(1): 198–203.

[32] Huang Decai and Guo Haidong, Hybrid Genetic Algorithm for Solving Job Scheduling Problem on Non-identical Parallel Machines Based on JIT Technique. *Computer Integrated Manufacturing Systems*, 2004, 10(3): 288–302.

[33] Hu Haiya and Ye Feifan. Ant Colony Algorithm Based Parallel Flow Shop Scheduling Method. *Mechanical Engineering*, 2005, 43(9): 26–28.

[34] George, C., and Velusamy, S. Dynamic Scheduling of Manufacturing Job Shops Using Genetic Algorithms. *Journal of Intelligent Manufacturing*, 2001, 12: 281–293.

[35] Pan Quanke and Zhu Jianying. Rolling Time Horizon Job Shop Scheduling Strategy in Dynamic Environment. *Journal of Nanjing University of Aeronautics & Astronautics*, 2005, 2(37): 261–267.

[36] Liang, W., and Yu, H. Learning Based Dynamic Approach to Job-Shop Scheduling. *Proceedings, Info-tech and Info-net, (ICII)*. Beijing. 2001, 3(11): 274–279.

[37] Shafaei, R., and Brunn, P. Workshop Scheduling Using Practical (Inaccurate) Data Part 1: The Performance of Heuristic Scheduling Rules in a Dynamic Job Shop Environment Using a Rolling Time Horizon Approach. *International Journal of Production Research*, 1999, 37(17): 3913–3925.

[38] Ovacik, M. I., and Uzsoy, R. Rolling Horizon Procedures for Dynamic Parallel Machine Scheduling with Sequence-Dependent Setup Times. *International Journal of Production Research*, 1995, 33(11): 3173–3192.

[39] Ovacik, M. I., and Uzsoy, R. Rolling Horizon Algorithms for a Single-Machine Dynamic Scheduling Problem with Sequence-Dependent Setup Times. *International Journal of Production Research*, 1994, 32(6): 1243–1263.

[40] Chen Wen, Wang Shilong, and Huang He. Research on Ant Colony Algorithm for Multi-Agent Based Workshop Dynamic Scheduling. *Modular Machine Tool and Automatic Manufacturing Technique*, 2004, (6): 56–59.

[41] Safonov, M. G. *Stability and Robustness of Multivariable Feedback Systems*. Cambridge, MA: MIT Press, 1980.

[42] Zames, G. Feedback and Optimal Sensitivity: Model Reference Transformations, Multiplicative Seminorms and Approximate Inverses. *IEEE Trans. Automation Control*, 1981, 26(2): 301–320.
[43] Doyle, J. C., and Stein, G. Multivariable Feedback Design: Concepts for a Classical/Modern Synthesis. *IEEE Trans Automat Control*, 1981, 26(1) : 4–16.
[44] Doyle, J. C. Analysis of Feedback Systems with Structured uncertainties. *IEEE Proc. Pt. D*, 1982, 129(6): 242–250.
[45] Li Shaoyuan, Xi Yugeng, Chen Zengqiang, and Yuan Zhuzhi. The New Progresses in Intelligent Control (I). *Control and Decision*, 2000, 15(1): 1–5.
[46] Li Shaoyuan, Xi Yugeng, Chen Zengqiang, and Yuan Zhuzhi. The New Progresses in Intelligent Control (II). *Control and Decision*. 2000, 15(1): 1–5.
[47] Xi Yugeng. Paying Attention to the Research on Methodologies in Control Science. *Acta Automatic Sinica*, 2002, 28: 85–91.
[48] Du Wen and Huang Chongchao. Genetic Algorithms For Bi-Level Programming. *Journal of Mathematics (PRC)*, 2005, 25(2).
[49] Shuang Bing and Gu Xingsheng. Order-Oriented Scheduling for Continuous Production Process with Due-Date Windows. *Journal of Central South University of Technology (Natural Science)*, 2003. 34(4): 420–423.
[50] Liu Xiaobing, Sun Yongli, Hao Yingguang, and Wang Yuchun, Model of Iron and Steel Enterprises Group Strategy Production Scheduling and Planning Based on Theory of Hierarchical Optimization. *Computer Integrated Manufacturing Systems*, 2007, 13(5): 833–838.
[51] Guo Haidong, Huang Decai, and Shen Liangzhong. Job Scheduling Problem of Maximizing Customers' Satisfaction Degree on Parallel Machines Based on JIT Technique. *Journal of Zhejiang University of Technology*, 2004, 32(1): 82–85.
[52] Bao Zhengqiang, Li Changyi, and Zhou Xin. A Knowledge Based Dynamic Scheduling Decision System. *China Mechanical Engineering*, 17(13): 1366–1370.
[53] Zhang Jie. Research on Shop Floor Control Model Research. Postdoctoral Research Reports, Huazhong University of Science and Technology, 1999.

[54] Zhou Wei and Jin Yihui, Multi-Agent System Based MES Design. *Proceedings of the World Congress on Intelligent Control and Automation*, 2002, June 10–14: 2655–2659.

[55] Genesereth, M. R., and Ketchet, S. P. Software Agents. *Communication of the ACM*, 1993, 37(7): 48–53.

[56] Dorigo, M., Colorni, A., Maniezzo, V., et al. Distributed Optimization by Ant Colonies. *Proceedings of the European Conference on Artificial Life*. Paris, 1991, 134–142.

[57] Hadeli, P. V. Multi-Agent Coordination and Control Using Stigmergy. *Computers in Industry*, 2004, (5): 75–96.

[58] Dorigo, M., Colorni, A., Maniezzo, V., et al. Distributed Optimization by Ant Colonies. *Proceedings of European Conference on Artificial Life*. Paris, 1991: 134–142.

6

Multi-Agent-Based Production Scheduling in Re-Entrant Manufacturing Systems

6.1 Introduction

The re-entrant manufacturing system is characterized by lengthy processing cycles, various products in process, unstable product mix and expensive machines. With the increasing competition in the global market, it is necessary to introduce a production scheduling technology with optimal performance in a large-scale system level in order to reduce manufacturing costs, shorten processing cycles, and ensure on-time delivery.[1] Therefore, with the aid of the characteristics of the Multi-Agent technology, which includes autonomy, collaboration and adaptive ability, a hierarchical adaptive production scheduling scheme in Multi-Agent re-entrant manufacturing systems is developed to conduct a push-and-pull combined production scheduling process. In the production scheduling phase, considering preventive machine maintenance, batch processing machines, single-lot processing machines and product process constraints, the collaborative requirements of production scheduling process between single-lot processing machines and batch processing machines must be satisfied. A hierarchical collaborative production scheduling method is proposed while satisfying the optimal condition. In the rescheduling phase, an adaptive rescheduling method is presented to optimize production profit and inventory cost simultaneously by analyzing the impacts of constraints on the machine pools scheduling process, which includes the maximum batch size of batch processing machines, machine breakdown, random product rework and delivery due dates of various products.

6.2 Production Scheduling in Re-Entrant Manufacturing Systems

The core requirement of the production scheduling process in re-entrant manufacturing systems is to optimally allocate resources to both the system layer and the machine layer, so as to complete the jobs released by the production plan in an optimal way as far as possible, The scheduling tasks involve decomposing production tasks released by the production plan, generating the production activity control, feeding back the completion status of the production activity control in accordance with production process information provided by the Multi-Agent production control system, and rescheduling production tasks when the production progress cannot satisfy the requirements.

6.2.1 Re-Entrant Manufacturing Systems

The main difference between a re-entrant manufacturing system and a traditional manufacturing system such as Job Shops and Flow Shops is in the characteristics of composition and production organization. It is clear that semiconductor manufacturing systems are typically re-entrant manufacturing systems. Hence, in this book, we will consider semiconductor manufacturing systems when re-entrant manufacturing systems are presented. A semiconductor manufacturing system contains processing machines and material handling equipment.

6.2.1.1 Processing Machines

In general, semiconductor processing machines can be classified as single-lot processing machines (SPM) and batch processing machines (BPM).

a) SPM

In a semiconductor manufacturing system, quantities of wafers are grouped into a standard container, which is called a lot. The wafers in a lot are all the same type of products and travel as a unit among machines. Each SPM is capable of performing manufacturing operations on many wafers at a time and processing wafers one by one. After all the wafers in a lot have been completed, this lot of wafers will leave the SPM together. SPM can be further classified into three classes: single lot – single piece processing machines, single lot – multiple pieces parallel processing machines and MPM (single lot – multiple pieces multi-chamber processing machines). MPM can be sub-classified as MPM_SC (multi-chamber processing machine with same chambers) and MPM_DC (multi-chamber processing machine with different chambers). In the former, wafers can be processed in any available chamber one by one. On the other hand, many wafers are loaded, and should be processed in several chambers successively according to its processing sequence. Typical SPM includes RTP machine, ion implanter, CVD machine, coating and exposure device in lithography process, and ion etcher, and so on.

b) BPM

Each BPM is capable of simultaneously performing manufacturing operations on multiple lots of wafers with the same processing sequence. All the wafers in multiple lots will be processed together, and they will leave the BPM simultaneously. BPM can be further classified into two classes. The first one is batch parallel processing machines (BPPM), in which multiple lots of wafers are processed in parallel machines. The other one is batch serial processing machines (BSPM), in which multiple lots of wafers are processed in serial machines. Typical BPM includes horizontal and vertical oxidation furnaces, metal-etching machines, wet etching machine and so on.

6.2.1.2 Material Handling Equipment

Material handling equipment mainly includes buffer and material handling equipment. The buffer is a temporary storage system with various types. The most common storage system is the automatic shelf, which can be further sub-classified as the Cartesian type and the carousel type. Another storage system is under track storage (UTS), which is able to improve the efficiency of the automated material handling system (AMHS). The material handling equipment in semiconductor manufacturing systems consists of five types of equipment: vertical lifters, overhead tracks with transport vehicles, rail-guided vehicles (RGVs), automated guided vehicles (AGVs), and conveyors. The material handling system to transmit wafers in a production unit is called intrabay AMHS, while the material handling system to transmit wafers among production units is called interbay AMHS. AGV is widely applied as flexible and intelligent material handling equipment.

According to the production organization, processes, and processing model of semiconductor manufacturing systems, the characteristics of re-entrant manufacturing systems are summarized as follows:

1) Re-Entrant flow
   The nature of semiconductor manufacturing systems is re-entrant. In a traditional manufacturing system such as Job Shop and flow shop, the re-entrant flow occurs either in individual processes or in a few rework processes, which belong to the local phenomenon. However, in a re-entrant manufacturing system, a part (e.g., a wafer) may revisit the same machine pool multiple times throughout the manufacturing process. At different stages in the re-entrant flow, a wafer has to compete with others to be processed in the same machine pool. In particular, two reasons make the production scheduling process in re-entrant manufacturing systems different from that in traditional manufacturing systems. First, the hierarchical structure of semiconductor products is so strong that each layer is processed in similar operation sequences with different materials or precision. Hence, the re-entrant flow is adopted to process the next layer of the same product. (Shown in Figure 6-1.) Second, since machines required in wafer fabrication are very expensive, there is need to introduce re-entrant flow to improve machine utilization. Because of re-entrant flow, wafers visit the same machine pool many times throughout the manufacturing process. This results in significant increase in the number of wafers to be scheduled in the same machine pool, which enlarges the solution space and to increase the complexity of the production scheduling process.
2) Mixed processing mode
   In general, in semiconductor manufacturing systems, 25 or 50 wafers are grouped into a standard container, called a lot. Lots of wafers are processed in SPM and BPM. One lot of wafers can be processed in a SPM at a time, while lots of wafers with the same serial number or the same name of the manufacturing process can be processed in a BPM at the same time. The maximum

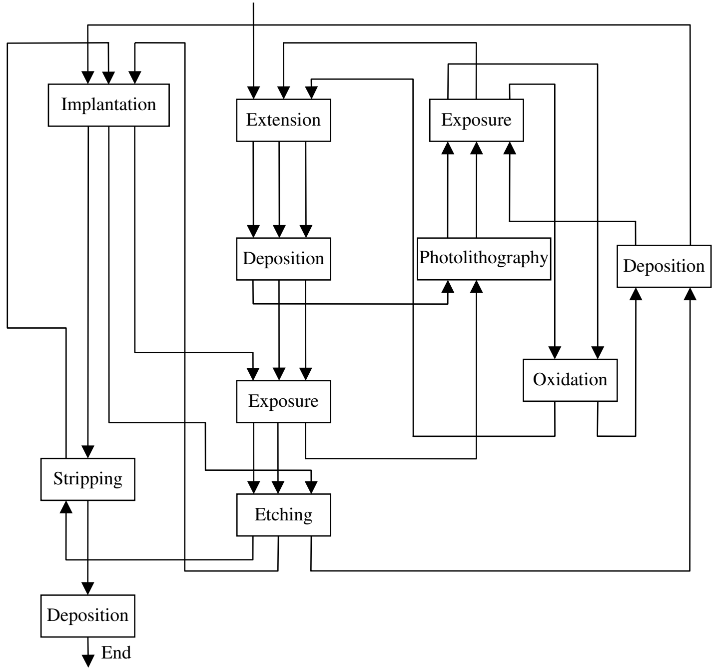

Figure 6-1 Front-end processing of semiconductor manufacturing.

batch size is regarded as an important parameter of BPM, which indicates its maximum processing capacity. As a major feature in semiconductor manufacturing systems, batch processing is an important factor needed to meet customers' demands for delivery due dates, and to affect the total production cycle of semiconductor production line. Considering differences between SPM and BPM, a different scheduling method should be proposed to properly arrange operation sequences in order to improve the productive efficiency of the total semiconductor production line.

6.2.2 Production Scheduling in Re-Entrant Manufacturing Systems

Semiconductor manufacturing is an industry with sophisticated technology, intensive funds and high-yield bonds, which is becoming a mainstay industry in some national economies. With rapid changes in market demands and intensified international competition, the procedure of production scheduling in semiconductor manufacturing systems has the following characteristics:

6.2.2.1 Large-Scale

In a typical wafer fab, there are dozens of processes and hundreds of machines. Each process consists of 300–900 processing steps processed in more than one hundred machines. The average production cycle time in the production line is approximately 30–60 days by repeatedly implementing manufacturing processes including oxidation, deposition, metallization, lithography, etching, ion implantation, photo-resist strip, cleaning, inspection and metrology. In addition, various products are processed in a semiconductor production line at the same time, which are different in thousands of operation sequences; even if similar products have different versions. This increases the complexity of semiconductor manufacturing processes. Furthermore, orders from different customers are processed in the semiconductor production line at the same time; there may be dozens of product types in these orders. Production scheduling problems in semiconductor manufacturing systems are complex due to the large product variation of semiconductor products.

6.2.2.2 Unbalanced Machine Workload

In semiconductor manufacturing systems, the workload of each machine is unbalanced. In particular, production efficiency is seriously restricted by complex processes with long processing times. Machines in semiconductor manufacturing systems are very expensive. Furthermore, these machines cannot be expanded in a linear way, that is to say, smoothly increasing machine capacity will directly increase production costs in a nonlinear way. Therefore, unlike many manufacturing systems, semiconductor manufacturing systems are developed on the basis of a predictable bottleneck. The performance of the whole system is determined by the production schedule that affects bottleneck machines. Hence, bottleneck machines should first be considered in the production scheduling process for semiconductor manufacturing systems. In a semiconductor production line, different products have different processes, and the unit operation processing time required in a machine may be so long that the machine becomes a bottleneck. However, longer processes of different products are not the same, which means that the bottleneck may shift from one machine to another. Consequently, the unbalanced machine workload would be more complex.

6.2.2.3 Dynamic Uncertainty

The semiconductor production line is a typical discrete dynamic system with a variety of uncertain events. These uncertain events can be classified into two categories: internal uncertain events and external uncertain events. Internal uncertain events are caused by internal problems such as machine breakdown, while external uncertain events are caused by changes in external conditions such as unstable orders. Uncertain events affect the running procedure of the production line, and damage the performance of the system.

a) Internal Uncertain Events

i. Machine Breakdown In semiconductor manufacturing systems, due to higher machine utilization, machines undergo serious stress so that their performance is gradually affected. Over time, decreasing performance could cause serious quality problems and reduce system reliability and reduce the yield rate. At the same time, machine breakdown in re-entrant Manufacturing Systems happens more frequently than in Job Shop or Flow Shop, which seriously affects system efficiency. Hence, it is necessary to introduce an efficient rescheduling strategy to improve the reliability of machines in order to achieve an effective production schedule.

ii. Trial run for New Products Semiconductor products update so fast that a trial run of new products is often required. However, due to the huge cost required for building a semiconductor production line, enterprises cannot develop a special trial production line for new products. Hence, products processed in the production line contain mass batch products and new products in the developing stage or in the small batch trial manufacturing stage. Sometimes these trial products are processed with priorities, which change the status of the production line and seriously affect system performance.

b) External Uncertain Events

i. Sudden Changes in Customers' Demand

Due to intensified international competition in the global semiconductor market, companies strive to satisfy customers' demands in order to obtain a larger market share. Therefore, a semiconductor production line should be able to respond rapidly to sudden changes in customers' demand such as order quantity and product variation. These uncertain events will affect the performance of a production line.

ii. Sudden Changes in Process In an actual semiconductor production line, according to the testing statistical data and customers' specific requirements, product engineers may temporarily add or change some processes in order to ensure performance and qualified rate of products. These sudden changes lead to change production parameters such as the operation processing time, which would greatly affect the system performance.

As the previous presentation shows, it is hard to implement production scheduling processes in semiconductor manufacturing systems due to their large scale and high uncertainty. It is necessary to introduce an efficient scheduling strategy to reduce work-in-process (WIP), shorten the production cycle time, and improve the overall equipment efficiency (OEE).

6.2.3 The Related Literature Review

The production scheduling and rescheduling problems in re-entrant manufacturing systems belong to the class of typical NP-hard problems. In the actual re-entrant manufacturing environment, for instance, long production cycle time, large product variation, process uncertainty, and shifting bottleneck, effective optimization methods are required to solve complex production scheduling problems in order to optimize production management procedures and to satisfy performance requirements of cycle time, delivery due date, output and WIP in re-entrant manufacturing systems. These approaches can be briefly classified into four categories:

6.2.3.1 Operation Research Methods

Operation research methods should solve various systematic optimization problems by using mathematical methods. Leachman[2] proposed a linear programming model for production planning and scheduling in semiconductor manufacturing systems in order to minimize the weighted production cost and to maximize machine utilization. Glassery[3] et al. used a linear programming model to obtain the feeding schedule for a single-variation and large-batch-size semiconductor production line. Sung[4] et al. developed a dynamic programming model to solve scheduling problems in semiconductor production lines with various part types and different arrival time for the parts. Dynamic programming algorithms are mainly applied in single-product single-batch equipment problems since they can ensure global convergence for small-scale problems. Sarin[5] et al. presented an integer programming model for production scheduling in a semiconductor production line so as to reduce cycle time. Sung and Choung[6] proposed an integer programming method to solve batch scheduling problems in semiconductor manufacturing systems. Gupta and Sivakumar[7] developed a multi-objective programming model based on multi-product semiconductor production lines in order to minimize the average production cycle time, to minimize the average tardiness, and to maximize machine utilization.

6.2.3.2 Heuristic Methods

The basic principle of heuristic rule is this: after an operation has been completed by using a machine, a job is selected from the waiting list in accordance with the heuristic rule. Kumar[8–11] et al. carried out much groundbreaking research for production scheduling problems in semiconductor production lines. In their papers, some common heuristic rules, including earliest due date (EDD), shortest processing time (SPT), and shortest least slack (SLK), were adopted to solve production scheduling problems in semiconductor manufacturing systems, and thus the performance of these scheduling rules were analyzed. Lu[12] et al. proposed three heuristic rules based on the concept of minimum relaxation: fluctuation smoothing policy of the average cycle time, fluctuation smoothing policy of the cycle time variances and fluctuation smoothing policy of tardiness. Qiao[13] et al. developed a new improved hybrid rescheduling strategy by comparing the existing periodic rescheduling policy, event-driven rescheduling strategy and hybrid rescheduling strategy. Hyung-Sik[14] adopted a neural network algorithm to solve rescheduling problems in re-entrant manufacturing systems. In this study, according to the system status, a dispatching rule was selected from a dispatching rule base by using a neural network algorithm in order to satisfy the requirements of dynamic scheduling.

6.2.3.3 Artificial Intelligence Methods

Artificial intelligence methods make a computer simulate a human brain to conduct engaged activities such as reasoning, programming, designing, thinking and learning, and so on.[15] Li[16, 17] et al. proposed a rescheduling strategy by analyzing triggered rescheduling factors in semiconductor manufacturing systems. In addition, they developed a rescheduling model based on swarm intelligence by further exploring the global modified rescheduling method. Savell[18] et al. used an expert system to implement production scheduling processes in semiconductor production lines. In their method, the expert system consisted of a central priority assignment module and a machine scheduling module. Liu[19] et al. proposed a method based on fuzzy logic to optimize the production scheduling process for lithography machines subject to specific constraints. Fargher[20] developed a fuzzy-logic-based planning and scheduling model for semiconductor manufacturing systems to deal with uncertain factors in the production cycle time in wafer fabrication. Huang[21] et al. presented a fuzzy neural network adaptive inference system to investigate real-time rescheduling decisions in re-entrant manufacturing systems.

6.2.3.4 Computational Intelligence Methods

The basic idea of computational intelligence methods is that a population of individuals who perform simple stress work collaborate by exchanging information in order to achieve high intelligence. The computational intelligent system provides many advantages over traditional systems: to name a few, strong self-organization, strong self-learning capabilities, high parallelism, robustness and nonlinearity. In contrast to traditional methods, with the aid of computational intelligence methods, it is more efficient to solve complex nonlinear problems. Applications of computational intelligence methods to complex nonlinear problems include pattern recognition, automatic knowledge acquisition, parallel search, associative memory, production scheduling and so on.[22] Qu[23] presented a tabu search algorithm for production scheduling problems in semiconductor manufacturing systems with a single machine in order to obtain approximate optimal solutions within an acceptable time. Geiger[24] formulated production scheduling problems at the wet etching process in wafer fabrication as flow shop sequencing problems so as to minimize makespan by considering material handling capacity, material processing constraints, and mixed storage policies. In addition, a tabu search algorithm was developed for this problem to obtain high-quality solutions in reasonable computational time. Bhushan[25] developed an algorithm that hybridizes a simulated annealing algorithm with a tabu search algorithm for scheduling problems in a wet etching unit with shared handling robots. Hiroyasu Toba[26] proposed a new segment-based real-time reactive rescheduling approach. In this study, a greedy scheduling algorithm was adopted to divide a wide scheduling range into small segments in order to improve production scheduling without decreasing production efficiency.

From the previous presentation, it is clear that optimization methods adopted for production scheduling processes in re-entrant manufacturing systems include operation research, heuristic methods, computational intelligence methods, artificial intelligence methods and so on. Scheduling methods based on operation research always assume that the system is stable, and these methods strive to eliminate the nonlinear characteristics of the system by linearization. These scheduling methods focus on “stability”, “balance” and “reasonable behavior”, which are not suitable for the dynamic environment of re-entrant manufacturing systems. Therefore, scheduling methods based on operation research are always applied in production planning. The local stage of the system is taken into consideration by scheduling methods based on heuristic rules. Nevertheless, these scheduling methods cannot achieve global optimization; this is their major shortcoming. The first limitation of computational intelligence algorithms is that their computational times are very long. The second limitation is that these scheduling algorithms cannot satisfy requirements of a real-time dynamic environment in re-entrant production lines such as large-scale, complex constraints and multi-objective optimization. Hence, practical application of such scheduling methods is rare. Although a Multi-Agent scheduling system is well openness, it is not sensitive to the scale of the problem, and it is able to achieve dynamic task assignment and production schedules,[27–30] implicit relevance among Agents and various relationships among tasks are neglected by current research, which affects system performance optimization. Therefore, in re-entrant, complex, dynamic scheduling environments of re-entrant manufacturing systems, Agent-based artificial intelligence methods are adopted to investigate hierarchical adaptive scheduling strategies for re-entrant manufacturing systems.

6.3 Multi-Agent-Based Hierarchical Adaptive Production Scheduling in Re-Entrant Manufacturing Systems

6.3.1 Hierarchical Adaptive Production Scheduling Strategy

The scheduling objectives of re-entrant manufacturing systems vary with the operation environment, processing tasks, types and quantity of manufacturing resources. There are numerous research projects on production scheduling models,[31] such as the NIST model,[32] Project 477,[33] Project 809,[34] the PCF method[35] and the DREAM method.[36] From previous studies, it is clear that the production management procedure of manufacturing systems includes releasing instructions, monitoring instructions and real-time responding to dynamic events.

In a re-entrant manufacturing system, due to large-scale machines, significant WIP inventory, long cycle time and many dynamic events, such as machine breakdown, process changes and defective products, the formulation of production schedules for such a system is a complex task. The production scheduling function of re-entrant manufacturing systems is expected to be superior to that of traditional manufacturing systems in terms of global optimal performance. In addition, its production scheduling function should be able to deal with major changes in the actual manufacturing environment, such as responding rapidly to sudden changes such as rush orders, random machine breakdown, preventive machine maintenance and so on. Besides, the monitoring function of re-entrant manufacturing systems should be extended to provide real-time information for all available manufacturing resources such as locations and processing status.

Therefore, the objectives are to formulate feasible production schedules for all manufacturing operations in re-entrant manufacturing systems, in order to respond quickly to sudden changes in tasks and the actual manufacturing environment, and to achieve hierarchy and adaptability as embodied by the production management procedure. In terms of the time-domain, the mechanism of production scheduling process can be presented as follows.

1) Production scheduling process in the system layer
   In the system layer, a multi-objective global scheduling mathematical model is developed to describe the characteristics of a re-entrant manufacturing system, which includes the constraints related to machine preventive maintenance, the maximum capacities of manufacturing resources such as BPM and SPM, and production recipes. The objectives are to formulate feasible production schedules for all manufacturing operations, in order to simultaneously optimize processing profits and the inventory cost.
2) Production scheduling process in the machine layer
   Production scheduling problems in the machine layer are characterized as a small-batch multi-objective scheduling problem with hundreds of processes, dozens of machines, and large product variation. There is need to optimize production schedules for both SPM pools and BPM pools in a large search space according to the real-time information provided by both up-stream and down-stream machine pools.
3) Rescheduling process
   In a dynamic, changeable and random manufacturing environment, it is necessary to consider dynamic events such as machine breakdown, lot rework and rush orders in re-entrant manufacturing systems. Therefore, a rescheduling strategy is adopted to respond rapidly to various dynamic events in order to improve system performance.

6.3.2 The Architecture of a Multi-Agent Hierarchical Adaptive Production Scheduling System

In contrast to traditional manufacturing systems, a semiconductor manufacturing system has the following special and complex features: re-entrant flow, co-existence of SPM and BPM, and unbalanced machine workload.[37–46] Considering the huge machine investment, significant WIP inventory, and large number of processes in re-entrant manufacturing systems, an efficient releasing and scheduling strategy is developed in order to improve machine utilization. Besides, considering various dynamic events in the manufacturing process (e.g., rush orders, random machine breakdown, wafer rework, and operator errors), a great deal of inaccurate information (e.g., the arrival time of each material, the unit operation processing time required in each machine, and the setup time of each machine) and incomplete information (e.g., quality of materials and quality of manufacturing parts), re-entrant manufacturing systems are required to effectively, accurately and rapidly respond to these factors in order to improve the efficiency and stability of their systems. Therefore, a Multi-Agent hierarchical adaptive production scheduling system is developed in this chapter, which is composed of production scheduling process in the system layer, production scheduling process in the machine/machine pool layer, and rescheduling process (shown in Figure 6-2).

Since the Multi-Agent hierarchical adaptive production scheduling system hybridizes advantages of a Multi-Agent federal system with those of a Multi-Agent hierarchical system, it reduces the network workload and improves system reliability. In a Multi-Agent system, the collaborative function of the system layer is different from that of the machine/machine pool layer, but they complement each other. The collaborative scheduling strategy in the system layer is constructed by using the collaborative scheduling Agent in combination with the task management Agent to form the central collaborative scheduling process in order to optimize production schedules in the system layer. The collaborative scheduling strategy in the machine/machine pool layer is generated by using the resource capacity management Agent in combination with a Multi-Agent production control system to form the collaborative mechanism in the machine tool layer so as to dynamically optimize local production

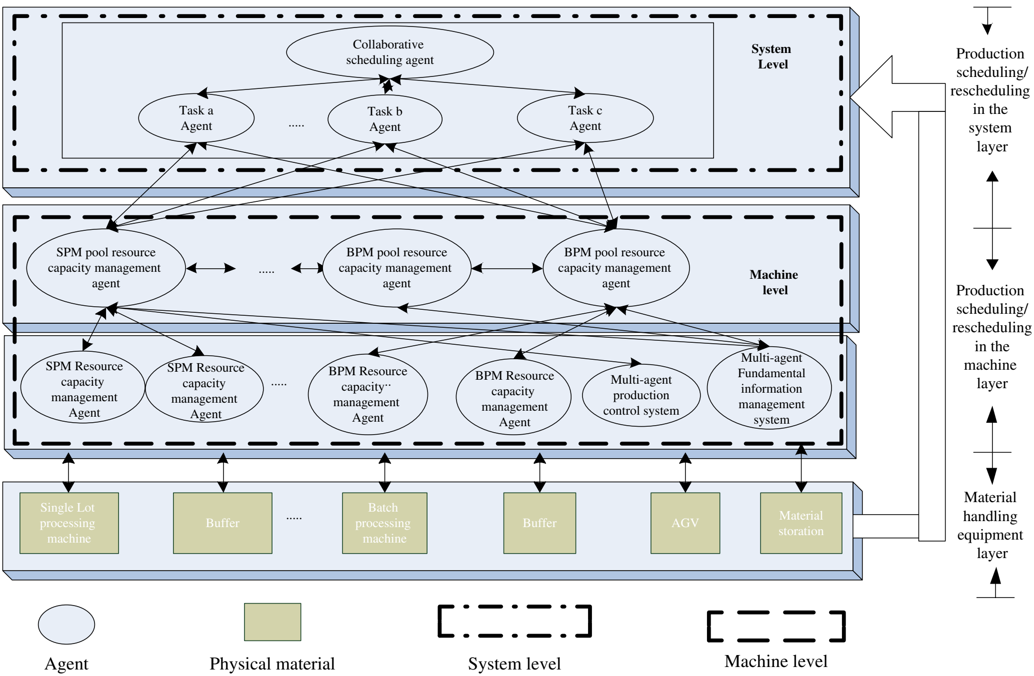

Figure 6-2 The architecture of Multi-Agent hierarchical adaptive production scheduling in re-entrant manufacturing systems.

schedules in the machine tool layer. The adaptive rescheduling strategy is implemented by using the collaborative scheduling Agent, the task management Agent, the resource capacity management Agent and a Multi-Agent production control system in order to maintain the stability of re-entrant manufacturing systems.

6.3.3 The Running Model for a Multi-Agent Hierarchical Adaptive Production Scheduling System

Figure 6-3 illustrates the mechanism of a Multi-Agent hierarchical adaptive production scheduling process in re-entrant manufacturing systems. Firstly, in the system layer, a short and fixed time interval (e.g., one day) is adopted to implement a production scheduling process in accordance with the rolling horizon procedure. Next, in the machine/machine pool layer, the lot moving plan is generated by the collaboration triggered by scheduling results of the system layer in order to optimize production schedules in the machine tool layer. Finally, when dynamic events occur, a hierarchical rescheduling process is performed according to the real-time status of the manufacturing system and the impacts of these events on the system. When the impacts are larger, a rescheduling process in the system layer is required for re-entrant manufacturing systems, and a rescheduling process in the machine layer is performed according to the scheduling results provided by the system layer; otherwise, there is only need to implement a rescheduling process in the machine tool layer.

6.4 Agents in a Multi-Agent Hierarchical Adaptive Production Scheduling System

Effective production schedules for re-entrant manufacturing systems are formulated by the task management Agent, the collaborative scheduling Agent and the resource capacity management Agent based on the collaboration of the Multi-Agent production control system and the Multi-Agent fundamental information management system. Hence, in this section, we will focus on introducing the architecture and behaviour models of

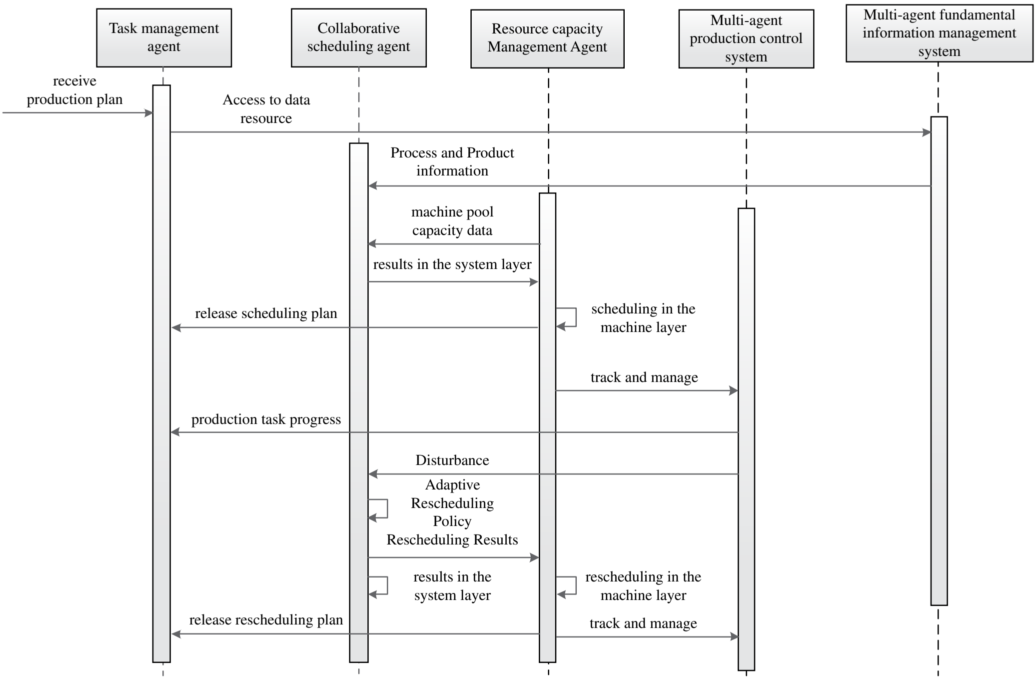

Figure 6-3 Hierarchical adaptive production scheduling in re-entrant manufacturing systems.

the task management Agent, the collaborative scheduling Agent, and the resource capacity management Agent in re-entrant manufacturing systems.

6.4.1 Task Management Agent

The task management Agent receives the tasks released by the Multi-Agent production planning system. When a task order arrives at a re-entrant manufacturing system according to the release plan, a specified task management Agent for this kind of task order is generated by a Multi-Agent production scheduling system. When the task has been completed, this task management Agent is cancelled by the Multi-Agent production scheduling system. The task management Agent is used to generate production schedules for tasks obtained by Lagrangian relaxation, and to record the following information: task machining process, task priority, task delivery due date and task release plan. The use case diagram of the task management Agent is illustrated in Figure 6-4.

The task management Agent is adopted to receive tasks assigned by the Multi-Agent production planning system, and to deliver tasks to the collaborative scheduling Agent, so as to collect the production processing information. In the whole process, simple reactive behaviors performed by the task management Agent are to collect and record the process information data, while thinking behaviour performed by the task management Agent should implement a complex calculation process to solve task sub-scheduling problems. Therefore, the

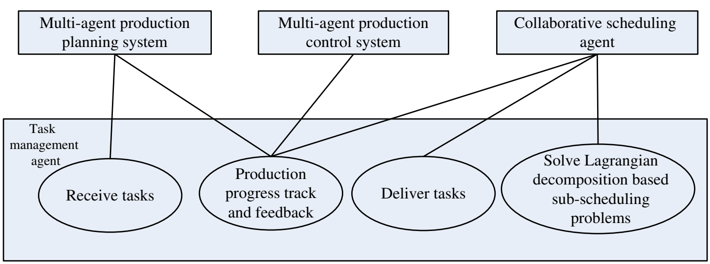

Figure 6-4 The use case diagram of a task management agent.

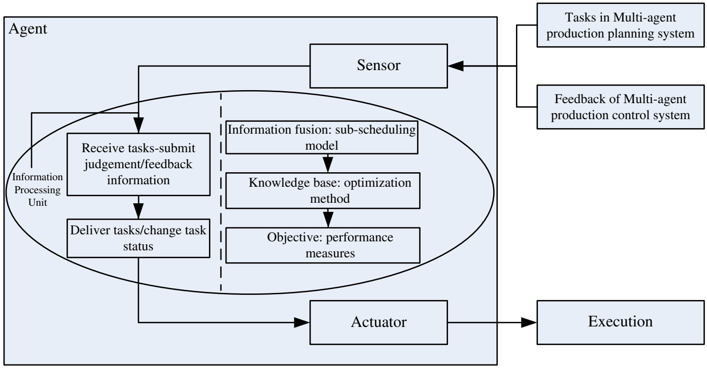

Figure 6-5 Structure of task management agent.

task management Agent in re-entrant manufacturing systems belongs to a mixed structure, shown in Figure-6-5.

6.4.2 Collaborative Scheduling Agent

In re-entrant manufacturing systems, the collaborative scheduling Agent is responsible for coordinating task management Agents according to capacity requirements of shared machine pools provided by each task management Agent, and for guiding scheduling results to improve the overall efficiency of the manufacturing system in order to obtain the suboptimal solution to the scheduling problem in the system layer. In the above process, the collaborative scheduling Agent is employed to exchange information with the task management Agent and the resource capacity management Agent. Figure 6-6 shows an illustration of

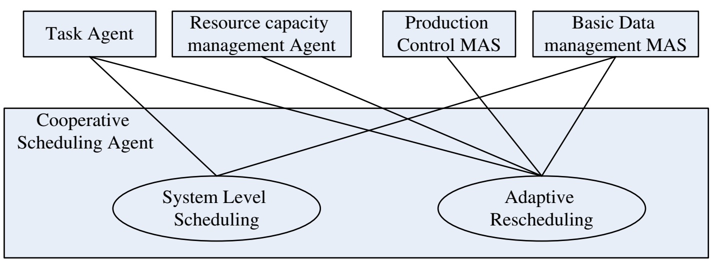

Figure 6-6 The use case diagram of a Collaborative Scheduling Agent.

the behaviour model of the collaborative scheduling Agent in re-entrant manufacturing systems.

In re-entrant manufacturing systems, the collaborative scheduling Agent is responsible for two schemes. First, at the production scheduling phase in the system layer, the collaborative scheduling Agent is adopted to configure basic information and parameters, and to calculate complex parameters to generate production plans satisfying constraints such as manufacturing processes, the delivery due dates of various products and the maximum capacities of manufacturing resources to maximize the resource utilization incurred in manufacturing the products. Next, in the adaptive rescheduling phase, the collaborative scheduling Agent is used to generate the rescheduling strategy by considering internal and external factors in the re-entrant manufacturing systems. Since a complex knowledge reasoning process is needed in these two phases, the collaborative scheduling Agent is regarded as a typical thinking Agent. Its mechanism is presented in Figure 6-7. In information processing, the information fusion model represents the production scheduling model in the system layer and the adaptive rescheduling model, the knowledge base is equivalent to scheduling and decision-making methods and the goal corresponds to various performance measures. The principle of the information processing unit will be described in detail in section 6.5 and 6.6.

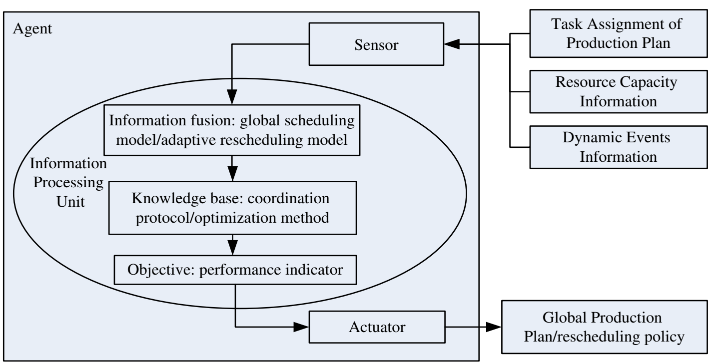

Figure 6-7 Mechanism of the Collaborative Scheduling Agent.

6.4.3 Resource Capacity Management Agent

It is known that SPM and BPM co-exist in re-entrant manufacturing systems. The resource capacity management Agent is further sub-classified as a resource capacity management Agent in the machine layer and a resource capacity management Agent in the machine pool layer in order to present the resource capacity management in re-entrant manufacturing systems in detail. That is to say, the resource capacity management Agent includes the SPM pool Agent, the BPM pool Agent, the SPM Agent and the BPM Agent. The machine/machine pool Agent is adopted to receive the production plan in the system layer released by the collaborative scheduling Agent in order to generate the production plan in the machine layer by further subdividing.

In re-entrant manufacturing systems, the resource capacity management Agent has two responsibilities. The first is to formulate production schedules in the machine layer, which belongs to a complex thinking behavior. Second, similar to the resource capacity management Agent in other manufacturing systems, it is mainly responsible for balancing resource capacity, recording resource capacity information and judging whether the results of production scheduling satisfy the resource capacity constraints, all of which belongs to a reaction behavior. Therefore, the resource capacity management Agent is regarded as a mixed Agent, and its mechanism is presented in Figure 6-8.

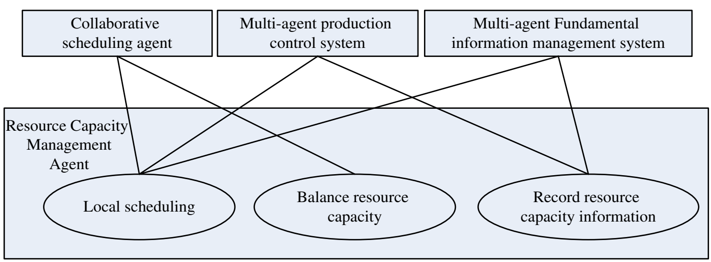

Figure 6-8 The use case diagram of a resource capacity management Agent.

6.5 Hierarchical Production Scheduling in Re-Entrant Manufacturing Systems

In re-entrant manufacturing systems, the hierarchical scheduling process is implemented by using the collaborative scheduling Agent in combination with the resource capacity management Agent. The collaborative scheduling Agent is adopted to generate feasible production schedules in the system layer, while the resource capacity management Agent is used to formulate feasible production schedules in the machine layer. In this section, according to the specific requirements of the information processing unit in the scheduling process, a knowledge base and a scheduling method are designed in order to optimize the production plan for re-entrant manufacturing systems, and to support internal coordination in a Multi-Agent scheduling system.

6.5.1 Problem Description

6.5.1.1 Production Scheduling in the System Layer

As TOC (Theory of Constraints) pointed out, during the lot processing, most waiting times happen in bottleneck machine pools. The bottleneck machine pools are key machine pools. Therefore, the system model is developed on the basis of key machine pools with limited capacity while non-key machine pools are treated as machines with unlimited capacity. The production scheduling process in the system layer focuses on performing capacity planning for key machines/machine pools. A processing process required to manufacture a part consists of a number of processing stages, and each processing stage contains a certain number of continuous processes. The first processing process is produced in the key machine pool, while others are produced in non-key machine pools. The operation processing time is simplified as fixed delay time, and the continuous time is discretized into several time periods in the planning horizon. Since the first processing process in every processing stage occupies the maximum capacity of the key machine pool, the operation processing time required in the first processing process must satisfy capacity constraints related to this key machine pool.

In order to facilitate the presentation, the following notation is used in the development of the mathematical model:

(i,s) – the processing stage s of part i, s=1,2,..,S_i, where S_i is the number of processing stages of part i;
m – key machine pool;
M_{is} – the key machine pool that is able to process processing stage s of part i;
u_{ist} – the number of processing stages of part i in period t;
X_{ist} – the number of processing stages for waiting part i at the beginning of period t;
r_{it} – the number of released part i at the beginning of period t;
t_{mt} – the maximum available capacity of key machine pool m in period t;
n_{is} – the number of periods required to manufacture processing stage s of part i;
t_{ist} – the capacity of the key machine pool required to manufacture processing stage s of part i in period t;
C_{is} – WIP inventory cost per period;
R_{is} – profit of processing stage s of part i.

In this case, the following assumptions are taken into consideration to construct the mathematical model:

a) The setup time of parts is fixed, and it is included in the processing time.
b) The capacity of buffers is infinite, and WIP inventory cost is neglected.
c) No rework parts and no defective parts.
d) BPM is simplified as SPM with a maximum capacity t_{mt}, which is defined as follows.

$$t_{mt} = t'_m f_m - \sum t_m \tag{6-1}$$

Where t'_m is the theoretical maximum capacity. f_m $(0 < f_m < 1)$ is the full batch coefficient according to the actual production statistics. t_m is the capacity occupied by preventive machine maintenance.

$$t'_m = B_m \cdot T_P \cdot N_m,$$

Where B_m is the maximum batch size of BPM. T_P is the length of a period. N_m is the number of machines in machine pool m.

Indeed, the mathematical model governing the hierarchical scheduling problem in the system layer has the following form:

$$\max \sum\sum\left(R_{is}u_{ist} - C_{is}X_{ist}\right) \tag{6-2}$$

$$\min \sum\sum\left(-R_{is}u_{ist} + C_{is}X_{ist}\right) \tag{6-3}$$

Subjected to:

$$X_{il(t+1)} = X_{ilt} - u_{ilt} + r_{it} \tag{6-4}$$

$$X_{is(t+1)} = X_{ist} - u_{ist} + u_{i(s-1)\left(t-n_{i(s-1)}\right)} \tag{6-5}$$

$$X_{i(s_i+1)(t+1)} = X_{i(s_i+1)t} + u_{iS_i\left(t-n_{iS_i}\right)} \tag{6-6}$$

$$\sum\sum u_{isj}t_{is(t-j+1)} \leq t_{mt} \tag{6-7}$$

$$u_{ist} \geq 0, X_{ist} \geq 0, u_{ist}, X_{ist} \in I \tag{6-8}$$

Where *(i, s_i+1)* is a virtual processing stage, which indicates that finished parts would ship to stock. $X_{i(s_i+1)t}$ is the number of finished parts in period *t*, and *I* is an integer set. Equation (6-2) is a multi-objective function, which aims to maximize the differences between total profits from all processing stages and total WIP inventory costs. This equation is equivalent to minimize objective function (6-3). Constraints (6-4) to (6-6) represent the inventory balance, which must satisfy the constraints related to the operation sequence. Constraints (6-7) represent the maximum capacity of key machine pools. Constraints (6-8) ensure that u_{is} and X_{ist} are positive integers.

(2) Production Scheduling in the Machine layer

The production scheduling process in the machine layer is to allocate tasks to machine pools.

In the mathematical model governing the hierarchical scheduling problem in the machine layer, a re-entrant manufacturing system consists of machine pools $S_{line}=\{G_1,G_2, \ldots G_i,\ldots,G_m\}$, where G_i represents the same type of machines, $G_i=\{e_{i1}, e_{i2}, \ldots e_{ip}\}$. T_{is} is the operation processing time required to manufacture processing stage *s* of part *i*, which contains the processing time to manufacture key processes T_{Kis}, and the processing time to manufacture non-key processes T_{Nis}, *i.e.* $T_{is}=T_{Kis}+T_{Nis}$.

In order to facilitate the presentation, the following notation is used in the development of the mathematical model:

t_{isk} – the starting time for manufacturing processing stage (i,s) of part L_k;
x_{kmnt} – a 0-1 binary decision variable where x_{kmnt} is equal to 1, if part L_k is assigned to machine e_{mn} in period t; otherwise, x_{kmnt} is equal to 0;
c_{mn} – if machine e_{mn} is a SPM, then c_{mn} = 1; if it is a BPM, c_{mn} is the maximum batch size of BPM;
R_{kmnt} – the process of part L_k that is assigned to machine e_{mn} in period t;
T_{Ph} – the length of a period;
T_{Pisk} – the production planning time for manufacturing processing stage (i, s) of part L_k.

Objective function:

$$\min \sum_{s}\sum_{t}\left[\left(\sum_{\substack{m \\ m=M_{is}}} \sum_{j=t-n_{is}+1}^{t} \lambda_{mj} t_{is(t-j+1)} - R_{is}\right) u_{ist+K_{is}X_{ist}}\right] - \sum_{m}\sum_{t} \lambda_{mt} t_{mt}$$

Subjected to:

$$\mathrm{t_{isk}} + \mathrm{T_{is}} \le \mathrm{t_{i(s+1)k}} \tag{6-9}$$

$$\sum x_{kmnt} \le c_{mn} \tag{6-10}$$

$$\sum\sum x_{kmnt} = 1 or 0 \tag{6-11}$$

$$T_{Ph}\left(T_{Pisk} - 1\right) \le t_{isk} \le T_{Ph} T_{Pisk} \tag{6-12}$$

Constraints (6-9) represent constraints related to operation sequences. Constraints (6-10) represent constraints related to the maximum capacity of a machine. Constraints (6-11) ensure that a processing stage of a part can only be assigned to one machine in period t. And if c_{mn}>1and x_{kmnt}=1,$k \in K$, then R_{amnt} = R_{bmnt}, a≠b, a, b$\in K$, where K is a set of waiting parts in the buffer of machines e_{mn}. Constraints (6-12) represent constraints related to the production planning time, which belong to a soft constraint.

6.5.2 Contact Net Protocol based Production Scheduling in the System Layer

In re-entrant manufacturing systems, the task management Agent acts as a tenderer, while the resource capacity management Agent acts as a bidder. In contrast, the collaborative scheduling Agent acts as an arbiter. First, in accordance with the capacity of machine pools required to manufacture waiting tasks for each part, the bidding documents are generated by the task management Agent. Second, the available capacity of machine pools is discretized into small periods as bidding objectives by the resource capacity management Agent. Finally, the optimal production schedule is generated by the resource capacity management Agent in order to bid in terms of the latest price and their respective objectives. The resource capacity management Agent is adopted to indirectly guide the decision of the tenderer by continuously adjusting the price of available periods of key machine pools in accordance with the supply and demand relationship of all the machine pools. The collaborative procedure of each Agent is illustrated in Figure 6-9.

6.5.2.1 Step 1) Initialize the Task Management Agent

According to the Lagrangian relaxation principle,[47] the collaborative scheduling Agent in re-entrant manufacturing systems is used to relax capacity constraints (6-7) of machine pools, and to transform the scheduling problem with multiple products and limited capacity into several sub-scheduling problems with single product and unlimited capacity, just as in (6-13) through (6-15). While constraints (6-3) are changed to become equation (6-16), regarding its corresponding single product scheduling problem, the task management Agent is created and instantiated according to the WIP product information obtained by resource capacity management Agent.

Capacity constraints of machine pools are relaxed as follows.

$$\min \sum_{(i,s)} \sum_{t} \left(-R_{is} u_{ist} + C_{is} X_{ist}\right) + \sum_{m} \sum_{t} \lambda_{mt} \left(\sum_{\substack{(i,s) \\ M_{is}=m}} \sum_{j=t-n_{is}+1}^{t} u_{isj} t_{is(t-j+1)} - t_{mt} \right) \tag{6-13}$$

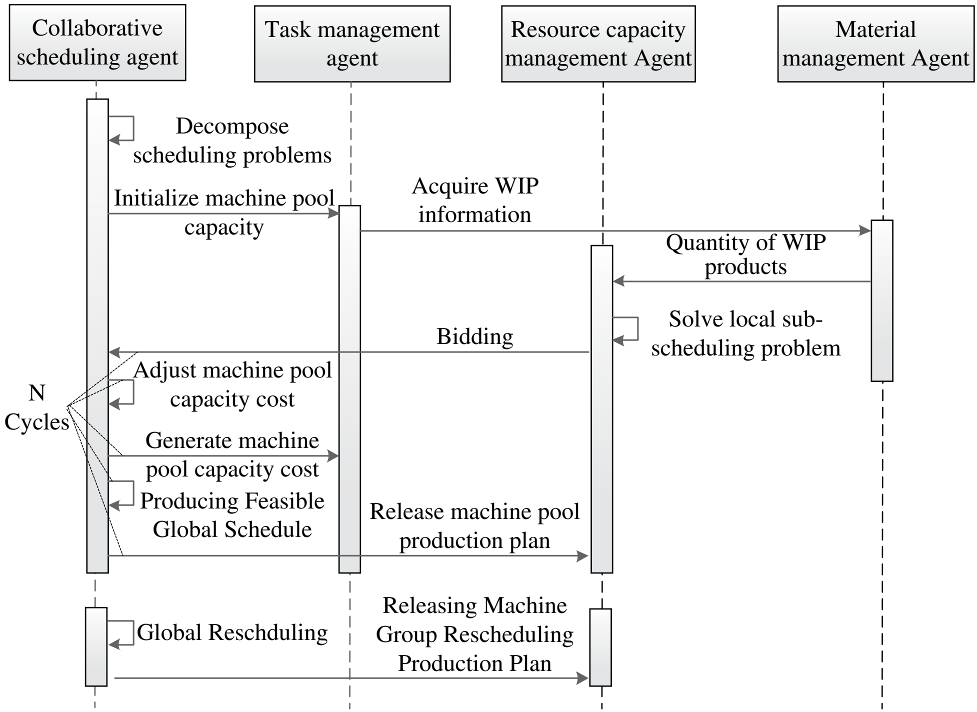

Figure 6-9 Contact net protocol based collaborative scheduling.

$$\min \sum_s \sum_t \left[\left(\sum_{\substack{m \\ m=M_{is}}} \sum_{j=t-n_{is}+1}^{t} \lambda_{mj} tis(t-j+1) - R_{is} \right) u_{ist} + K_{is} X_{ist} \right] - \sum_m \sum_t \lambda_{mt} t_{mt} \quad (6\text{-}14)$$

Where λ_{mt} is the corresponding Lagrangian multiplier of machine pool m in period t. The sub-scheduling problem PL_i, of product i is as follows.

$$PL_i(\mathbf{u}_i, \lambda) \equiv \sum_s \sum_t \left[\left(\sum_{\substack{m \\ m=M_{is}}} \sum_{j=t-n_{is}+1}^{t} \lambda_{mj} t_{is(t-j+1)} - R_{is} \right) u_{ist} + C_{is} X_{ist} \right] \quad (6\text{-}15)$$

Subjected to constraints (6-4), constraints (6-5), and constraints (6-6).

The corresponding dual problem is as follows.

$$\max q(\lambda) \equiv \sum_i \min_{\mathbf{u}_i} PL_i(\mathbf{u}_i, \lambda) - \sum_m \sum_t \lambda_{mt} t_{mt} \qquad \lambda_{mt} \geq 0 \quad (6\text{-}16)$$

6.5.2.2 Step 2) Perform the Scheduling Strategy by the Task Management Agent

As inventory balance constraints (6-1) through (6-7) pointed out, the sub-scheduling problem with single product and unlimited capacity (6-11) belongs to the minimum cost network flow problem. As the developed network regarding this problem, shown in Figure 6-10, node n_{ist} denotes the buffer of the processing stage (i,s) in period t. After n_{is} periods, the processing stage (i,s) has been completed and will be transformed to the processing stage $(i,s+1)$ to generate the processing flow u_{ist}, which is denoted by edge $[n_{ist}, n_{i(s+1)(t+n_{is})}]$ with upper bound t_{mt}. The unproduced processing stage (i,s) would still remain in the buffer to generate the buffer flow X_{ist}, which is denoted by edge $[n_{ist}, n_{is(t+1)}]$. Flow X_{is1} enters node n_{is1}, which represents the WIP part level in the buffer of each processing stage in the first period. Flow $u_{is}(k)(k = 0, 1, \ldots n_{is} - 1)$ enters node $n_{is(nis-k)}$, which represents the cross-period processing flow to enter the current planning period from the former one. Flow r_{it} enters node n_{i1t}, which represents the tasks obtained by the release plan to be

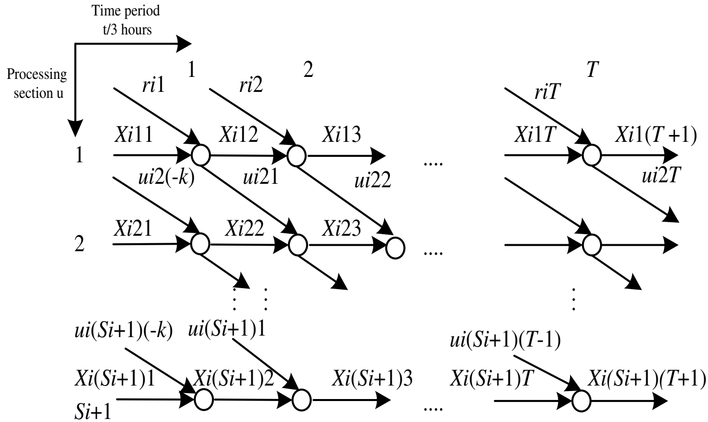

Figure 6-10 Network flow.

produced in the re-entrant manufacturing system. $(i,si+1)$ is a virtual processing stage. Flow $u_{is_i(t-n_{is_i})}$ emerges from node $n_{i(s_i+1)t}$, which represents completed tasks. The cost of edge $[n_{ist}, n_{i(s+1)(t+n_{is})}]$ is presented as follows.

$$\sum_{\substack{m \\ m=M_{is}}} \sum_{j=t-n_{is}+1}^{t} \lambda_{mj} - R_{is} \tag{6-17}$$

The cost of edge $[n_{i\,s\,t}, n_{i\,s(t+1)}]$ is K_{is}. The RELAXIV algorithm is adopted by the task management Agent to solve the sub-scheduling problem with single product and infinite capacity, and then to send bidding documents to the collaborative scheduling Agent in the re-entrant manufacturing systems. Each bidding document corresponds to a sub-scheduling plan.

6.5.2.3 Step 3) Adjust the Price of the Machine Pool in Each Period by the Resource Capacity Management Agent

According to the support and demand relationship of machine pools and equations (6-18) and (6-19), the collaborative scheduling Agent is used to collect all the bidding documents in order to calculate the length of each step, and to adjust the price of machine pools (per period) in the system level by adjusting the Lagrangian multiplier shown in equation (6-20).

Subgradient:

$$g_{mt}\left(\boldsymbol{u}^{*}\right) \equiv \frac{\partial q(\lambda)}{\partial \lambda_{mt}} = \sum_{\substack{(i,s) \\ M_{is}=m}} \sum_{j=t-n_{is}+1}^{t} u_{ist} - t_{mt} \tag{6-18}$$

The length of each step:

$$\theta_k = \beta \frac{q' - q(\lambda_k)}{g(\mu_k)^2} \tag{6-19}$$

Where q' is the estimated value of optimal dual cost.

The Lagrangian multiplier:

$$\max\{0, \lambda_k + \theta g(\boldsymbol{u}_k)\} \tag{6-20}$$

6.5.2.4 Step 4) Adjust Production Schedules in the System Layer

The capacity auction of machine pools is accomplished by several rounds of bidding from step 2) to step 3). Due to the existence of integer decision variables, the dual equation (6-16) may generate solutions that cannot satisfy constraints (6-7); it is necessary to adjust the production schedule in the system layer to make it feasible. In the whole scheduling process, the collaborative scheduling Agent is increased as periods in order to check whether all the machine pools satisfy constraints (6-7) in period $[t_k, t_{k+1}](1 \le t \le T)$. Regarding the machine pool that cannot meet the requirements, tasks u'_{ist} will be removed from edge $[n_{ist}, n_{i(s+1)(t+n_{is})}]$ with the least priority one by one, until they satisfy capacity constraints of machine pools. In order to maintain inventory balance constraints (6-4) through (6-6), removed tasks u'_{ist} will be added to the buffer flow X_{isk} $(t+1 \le k \le T)$. Meanwhile, tasks u'_{ist} are required to be eliminated from the downstream sub-network of edge $[n_{ist}, n_{i(s+1)(t+n_{is})}]$ in accordance with the RELAXIV algorithm in step 2).

6.5.3 GPGP-CN Protocol Based Production Scheduling in the Machine Layer

In the production scheduling process in the machine layer for re-entrant manufacturing systems, SPM and BPM couple with each other. Although there are five basic collaborative mechanisms based on Generalized Partial Global Planning (GPGP), it

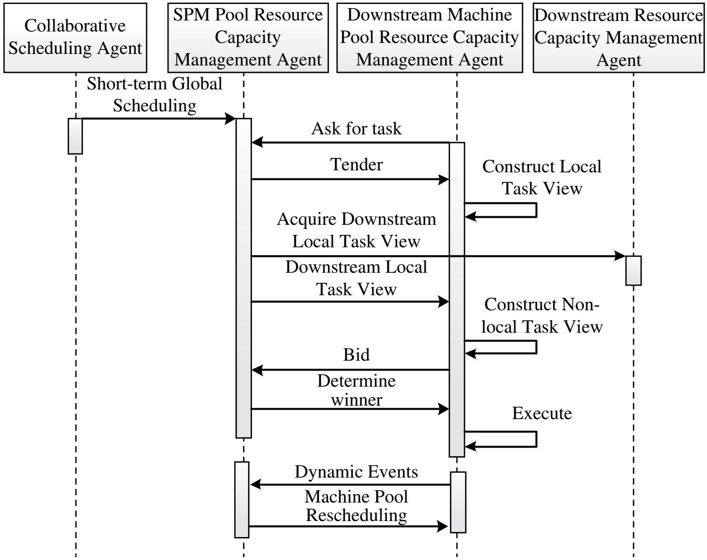

Figure 6-11 GPGP-CN based SPM pool Collaborative Scheduling.

is hard to implement them to solve such complex scheduling problems. In order to overcome this drawback, in this section it is necessary to introduce an approach called GPGP-CN based collaborative mechanism. This collaborative mechanism consists of two portions: a dynamic collaborative mechanism in SPM pool and a dynamic coordination mechanism in BPM pool, which are illustrated in Figure 6-11 and Figure 6-14, respectively. The GPGP-CN based collaborative mechanism is proposed to obtain upstream and downstream collaborative information by GPGP in combination with the contract net protocol–based bidding mechanism to implement production scheduling process in the machine layer for re-entrant manufacturing systems by cooperation and competition among Agents.

6.5.3.1 Production Scheduling Method in SPM Pool

(1) Tendering

The resource capacity management Agent receives the task production plan $\{P_{kt}|t=1,2,\ldots,M\}$ released by the collaborative scheduling Agent, where P_{kt} is the processing task set of machine pool G_k in period t, and M is the maximum planning period.

And then the Agent of machine pool G_k generates two processing task sets: waiting tendering task set W_k and tendering task set S_k.

$$W_k = B_{kj} \text{ if } t = j$$

where B_{kj} is the WIP product set in the buffer of machine pool G_k in period j.

$$\begin{cases} S_k = B_{kj} \cap (P_{kt} \cup L_{sk}) & \text{if} \quad t = j \quad \text{and} \quad \mathrm{B_{kj}} \cap (P_{kt} \cup L_{sk}) \neq \phi \\ S_k = B_{kj} & \text{otherwise} \end{cases}$$

where

$$t = \frac{j}{T_{ph} 60} \tag{6-21}$$

Ls_k is the tardy task set that cannot be processed in the planning period.

The resource capacity management Agent is adopted to trigger the SPM pool to tender by using three events, that is, SPM idling, BPM asking the resource capacity management Agent to implement collaborative scheduling, and SPM pool rescheduling caused by machine breakdown. The SPM pool resource capacity management Agent issues the tendering task set S_k to all machines in the pool.

(2) Collaborating

Each SPM resource capacity management Agent develops the ETAEMS-based local task view shown in Figure 6-12 in accordance with the tendering task set S_k., and checks collaborative requirements of local scheduling decisions. In order to maintain WIP products in a uniform distribution, the machine required to manufacture the key process with least WIP in the SPM pool would be selected with the priority. Each SPM resource capacity management Agent asks the resource capacity management Agent in re-entrant manufacturing systems to collaborate. The system resource capacity management Agent will determine the key machine pool required to manufacture the next processing stage of the selected job, and send the WIP information of this machine pool to the upstream SPM resource capacity management Agent.

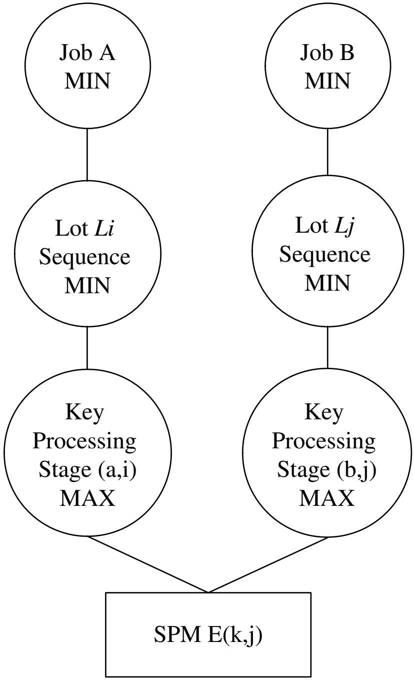

Figure 6-12 Local task view of each SPM Resource Capacity Management Agent.

(3) Bidding

Each SPM resource capacity management Agent constructs the ETAEMS-based non-local task view shown in Figure 6-13 according to collaborative information and the tendering task set S_k.

First, the earliest starting time t to manufacture the tendering task set is decided. Next, the task to be processed in period t is determined according to equation (6-22) and equation (6-23).

Indeed, the mathematical model governing the SPM production scheduling problem has the following form:

$$\max g\left(\underset{L_{Tk}=(i,s)}{w, L_k}\right) \tag{6-22}$$

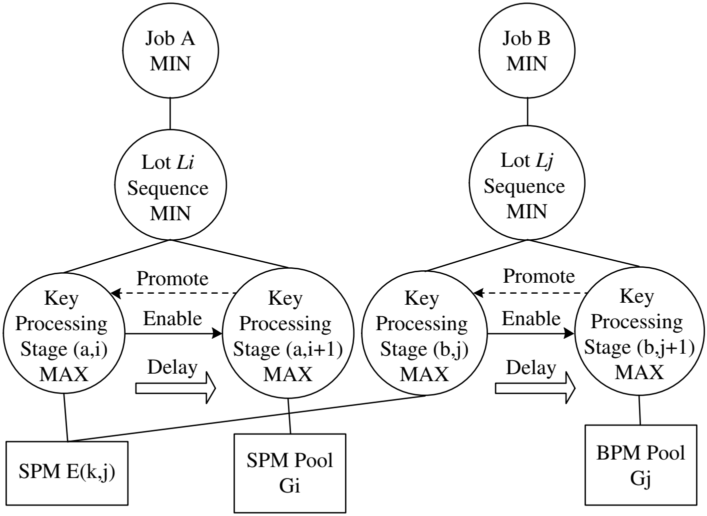

Figure 6-13 Non-Local Task View of each SPM Resource Capacity Management Agent.

$$g\left(\underset{L_{Tk}=(i,s)}{w, L_k}\right) = \left(w_1 A_{is} - w_2 \frac{\sum_{\substack{(i,j)\\ M_{ij}=M_{i(s+1)}}} T_{Kij} X'_{ijt}}{C_{M_{i(s+1)}}} - w_3 \frac{t_{Dk} - t_{Dks}}{\sum_{j=s+1}^{S_i} T_{Kij}} \right) \tag{6-23}$$

In order to facilitate the presentation, the following notation is used in the development of the mathematical model:

L_{Tk} – current processing stage of task L_k;
(i,s) – processing stage s of task type i, $s=1,2,..,S_i$, where S_i is the total number of processing stages of task type i;
w_1, w_2, w_3 – weighted coefficient;
A_{is} – priority value of processing stage (i,s);
M_{ij} – the machine pool to manufacture processing stage (i,s);
X'_{ijt} – the number of waiting processing stages (i,j) in the buffer of machine pools to manufacture the next processing stage in period t. The number of waiting processing stages (i,j) in the buffer in period k can be taken into consideration as the approximate value.

$$X'_{ijt} = \begin{cases} X'_{ijk}, & T_{Ck} \cdot T_{Ph} \cdot 60 \le t < \left(T_{Ck}+1\right) \cdot T_{Ph} \cdot 60 \\ X_{ijf}, & t \ge \left(T_{Ck}+1\right) \cdot T_{Ph} \cdot 60 \end{cases} \tag{6-24}$$

Where

$$T_{Ck} = \left[\frac{k}{T_{Ph} \cdot 60}\right], f = \left[\frac{t}{T_{Ph} \cdot 60}\right] \tag{6-25}$$

X_{ijf} – the planning amount of waiting processing stage (i,s) in the buffer of the machine pool to manufacture the next processing stage at the beginning of period f in the system layer short-term collaborative scheduling;

T_{Ck} – the current period k (minutes);

T_{ph} – the length of a period (hours);

C_{Mij} – available capacity of the key machine pool to manufacture processing stage (i,s) (hours);

t_{Dl} – the delivery due date of task L_l;

t_{Dls} – the earliest completion time for processing stage s of task L_l.

In equation (6-23), the first part represents the priority value of processing stage (i,s), while the second part represents the relative WIP level of the machine pool to manufacture the next processing stage. Due to the small fluctuations of the relative WIP level, the relative WIP level X'_{ijk} in current period k and the planning WIP level X_{ijf} at the beginning of period f determined by the system layer short-term collaborative scheduling are adopted instead of using X'_{ijt}. The third part represents the relative remaining processing time for processing current part. Each SPM resource capacity management Agent determines its own weighted coefficient. The bidding information of SPM resource capacity management Agent includes task ID, earliest completion time for processing each task, and workload level of machine e_{ij} in current period r, which is defined as follows.

$$U_{eij} = \sum_{\substack{(i,s) \\ M_{is}=e_{ij}}} \sum_{t \in [0,r]} u'_{ist} T_{kis} \tag{6-26}$$

Where e_{ij} is the machine able to manufacture processing stage (i,j); u'_{ijt} is the amount of processing stage (i,j) to be produced in period t.

(4) Contract Signing

After receiving the bidding documents from all the SPM resource capacity management Agents, the SPM pool resource capacity management Agent determines the bidding winner according to the decision-making rule, which includes ECT rule (Select the machine with the earliest Completion Time) and LWL rule (Select the machine with the Lowest Workload Level). The procedure of bidding is presented below:

Step (1): the ECT rule is used to select the bidding winner.

Step (2): if there are multiple choices, then the LWL rule is taken into consideration.

Step (3): if there are still several candidates, then randomly select one machine.

Step (4): after having decided the bidding winner, the signing information is issued to the selected SPM resource capacity management Agent. The signed task is added to the waiting task list by the winning SPM resource capacity management Agent.

Iteratively implement step (1) to step (4) many times until the tendering job set S_k is empty.

(5) Implementing the Signed Task and Rescheduling in Machine Pool

In the procedure of implementing the signed task, if machine breakdown is detected by the process monitoring Agent, the SPM pool resource capacity management Agent will be informed to renew the machine status information. In addition, the SPM pool resource capacity management Agent is adopted to add the waiting tasks produced in the breakdown machine into the tendering job set S_k, and to request the SPM pool resource capacity management Agent to tender. Besides, the SPM pool resource capacity management Agent is employed to deliver the changed results to the downstream SPM resource capacity management Agent in order to make corresponding adjustments. After the signed task has been completed, this information is issued from the SPM resource capacity management Agent to the resource capacity management Agent. The resource capacity management Agent will update the task processing information, and then detect

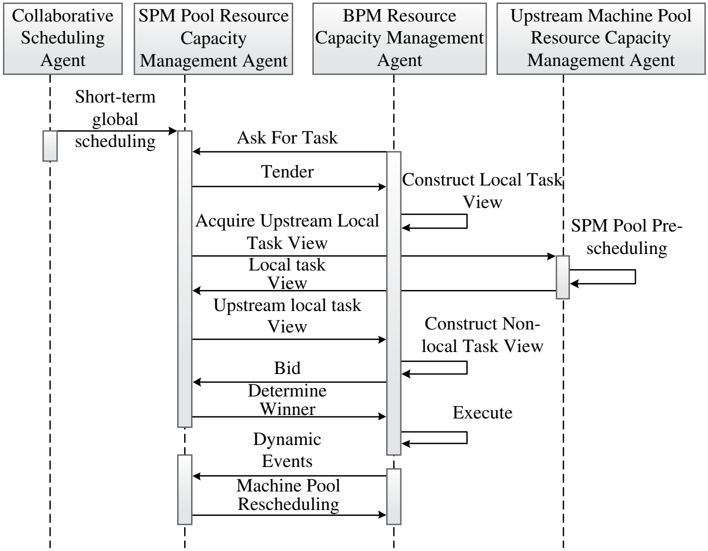

Figure 6-14 GPGP-CN based BPM group Collaborative Scheduling.

whether there are idle machines. If there are idle machines, then SPM resource capacity management Agent will request the resource capacity management Agent to tender. In addition, the signed task should have completed the subsequent non-key operations before it enters the waiting tendering task set W_j produced in the key machine pool G_j.

6.5.3.2 Production Scheduling Method in BPM pool

(1) Tendering

The tendering process of BPM is similar to that of SPM except for two points. One is that the triggering events are rescheduling caused by BPM idle and machine breakdown. The other one is the tendering information, which contains task ID and task processing time.

(2) Collaborating

Each BPM resource capacity management Agent constructs an ETAEMS-based local task view as shown in Figure 6-15, and checks collaborative requirements of local scheduling decision. The batch size and starting processing time of BPM are determined

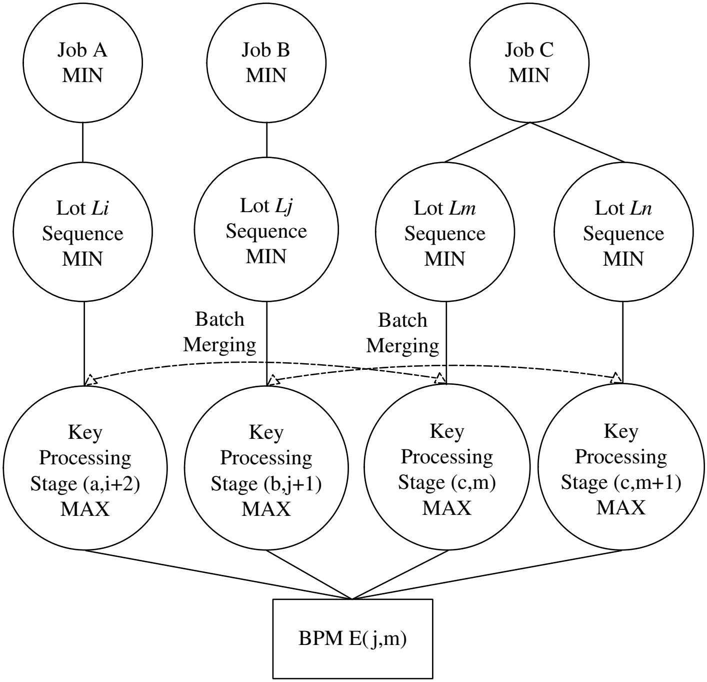

Figure 6-15 Local task view of each BPM Resource Capacity Management Agent.

by considering the arrival time of upstream tasks with the same recipe as the tasks in tendering task set S_k. And this leads to a batch-merging collaborative relationship between SPM production scheduling and BPM production scheduling.

The BPM resource capacity management Agent requestS the resource capacity management Agent to collaborate in order to obtain the arrival time of the corresponding upstream tasks in period $[t,t+T_{Kis}]$, where t is the tendering triggering time, and T_{Kis} is the processing time of the tendering task (i, s). Then the BPM pool resource capacity management Agent gathers all the collaborative demands according to the system layer short-term collaborative production schedule.

$$\bigcup_{(i,j)\in S_k} \left[t, t + T_{Kij}\right] \tag{6-27}$$

The BPM resource capacity management Agent requests the SPM pool resource capacity management Agent to collaborate in order to determine the SPM pool with a batch-merging collaborative relationship in accordance with the arrival time of corresponding upstream jobs and the ETAEMS-based global task view shown in Figure 6-12.

The SPM pool resource capacity management Agent determines the arrival time of upstream tasks to the BPM pool due to the collaborative scheduling mechanism.

Regarding tasks with several operations before entering BPM pool, firstly, the completion time of the key process is determined by tendering in the current machine pool; and the subsequent non-key processes are simplified as ones with fixed delay time. Then the arrival time of the task is sent to the next resource capacity management Agent to make a tender. Repeat this process until entering the BPM group. Finally, the arrival time of upstream tasks to enter the BPM pool is sent to each SPM resource capacity management Agent.

(3) Bidding

Each BPM resource capacity management Agent constructs the ETAEMS-based non-local task view shown in Figure 6-16 according to the collaborative information, and determines the tasks that can be batch-merged according to batch-merging constraints. The task with the maximum batch size will be selected with the priority, and will be started to process in idle period t_0. If there are multiple tasks with the maximum batch size, the priority level will be used to select. If there are still several candidates, randomly select one. However, if there is no task with the maximum batch size, the waiting cost of each task is calculated while starting the process in idle period t_0 according to equation (6-27). The BPM resource capacity management Agent bids to the tasks with the minimum waiting cost according to equation (6-28), and determines the starting time for processing according to (6-29).

Indeed, the mathematical model governing the BPM production scheduling problem has the following form:

$$\min_{t_0 < t_1, t_2, \cdots t_n < t_0 + T_{Pr}} g(t_{r1}, t_{r2}, \cdots t_{rn}, r) \tag{6-28}$$

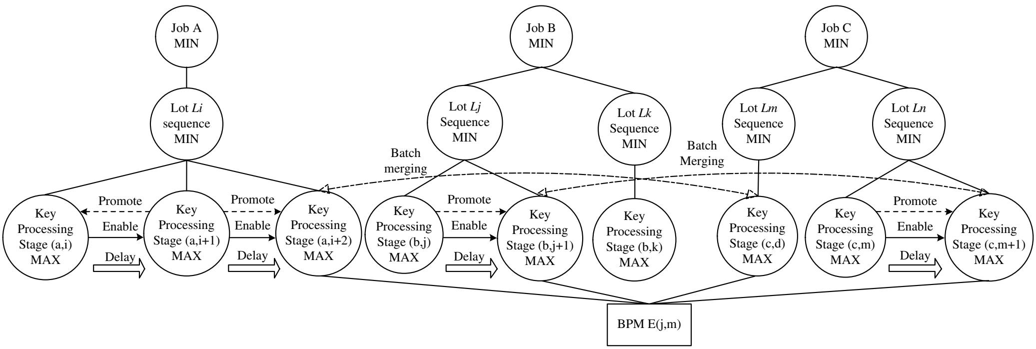

Figure 6-16 Non-local task view of each BPM Resource Capacity Management Agent.

$$g\left(\underset{t_0<t_1,t_2,\cdots t_n<t_0+T_{Pr}}{t_{r1},t_{r2},\cdots t_{rn}},r\right)=\frac{C_r}{\min\left(q_r,B_r\right)} \tag{6-29}$$

Where

$$C_r=T_{Pr}\left[\max\left(q_r-B_r,0\right)+\sum_{i\neq r}^{N}q_i\right]+\sum_{(i,j)}\sum_{t_0<t<t_0+T_{Pr}}\left[n_{ijt}\left(t_0+T_{pr}-t\right)\right] \tag{6-30}$$

In order to facilitate the presentation, the following notation is used in the development of the mathematical model:

t_{ri} – the arrival time of the ith job at the BPM pool in period $[t_0, t_0+T_{Pr}]$.
r – process ID of processing stage (i, j). As long as process ID is the same, even if the product type is different, products could be merged to one batch.
C_r – the total waiting cost of all tasks manufactured in process r.
q_r – the quantity of tasks (in tendering job set S_k) manufactured in process r in period t_0.
B_r – the maximum batch size of tasks manufactured in process r.
T_{Pr} – the processing time for processing tasks manufactured in process r (hours).
n_{ijt} – the quantity of new arriving tasks at the BPM pool in period t.

In equation (6-29), the first part represents the WIP waiting cost in period $[t_0, t_0+T_{Pr}]$ without considering new arriving tasks, while the second part represents the WIP waiting cost with new arriving tasks in this period.

$$t=\underset{t_i\in[t_0,t_0+t']}{\arg\max}\left\{\sum_{\substack{(i,j)\\R_{ij}=r}}\sum_{t\in[t_0,t_i]}n_{ijt}\left(t_0+T_{Pr}-t_i\right)-q_r\left(t_i-t_0\right)\right\} \tag{6-31}$$

Where $t'=\min\{T_{Pr}, t_{rc}\}$.

t_{rc} – the time to reach the maximum batch size for tasks manufactured in process r in the buffer of BPM pool.

t_i – the arrival time of the ith batch of tasks while considering the batch merging.
R_{ij} – the process of processing stage (i,j).

Equation (6-30) differentiates the WIP waiting cost processed in BPM in period t and the WIP waiting cost processed in BPM in period t_0. Its first part represents the saved WIP waiting cost in period $[t_0, t_0+T_{Pr}]$ when the task starts to be processed in BPM in period t. The second part represents the increased WIP waiting cost in period $[t_0, t_0+T_{Pr}]$.

(4) Contract Signing, Implementing the Signed Task, and Rescheduling in the Machine Pool

The procedures of BPM which consists of contract signing, implementing and machine pool rescheduling are similar to those of SPM except for the bidding process. In this case, that is to iteratively implement step (1) to step (4) of the bidding process many times until all the BPMs are occupied or the tendering job set S_k is empty. The non-bidded BPM resource capacity management Agent should remove the signed task decided by the collaborative decision with the SPM pool resource capacity management Agent from its waiting task list. While accepting the updated results provided by the upstream SPM rescheduling process, the BPM resource capacity management Agent should determine the starting time for batch processing according to equation (6-30).

6.5.4 Case Study

In this section, the combined auction-based hierarchical scheduling method is proposed for solving the hierarchical scheduling problem for re-entrant manufacturing systems. An Intel PIII–based computer is used to run the software package developed by using C++Builder5. Each Agent corresponds to a software object, and functions required in collaborative procedure are realized by functions of each object.

The practical data collected from a six-inch semiconductor fabrication line (SFL) in Shanghai is used to demonstrate the effectiveness of the proposed hierarchical scheduling algorithm. This SFL is composed of 11 key machine pools, which add up to 34 key machines with MTTF and MTTR parameters, as shown in Table 6-1. The full batch coefficient f_m is defined as 0.6 according to the practical statistical data. The capacity of the machine pool occupied by preventive maintenance activities D_m is randomly defined before implementing the production scheduling process without considering the rescheduling process. There are three kinds of lot product: "A", "B", and "C". The whole manufacturing procedure is divided into 85 processing stages shown in Table 6-2.

In this case, three kinds of wafer are released in the proportion 1:1:1. The production line is not empty at the beginning. That is to say, pre-occupied tasks have been allocated to the machines. The planning horizon is one day, and the length of a period is three hours. The production scheduling method in the system layer proposed in this book is applied to formulate production schedules in this case study. The FCFS (first come, first served) rule is used for SPM as a machine scheduling strategy in practice; whilst the MBS (Minimize batch size) rule is adopted for BPM as a machine pool scheduling strategy in practice. The minimum batch size is 2 minutes. The experiment is carried out for a week, and the results are summarized in Table 6-3. In the collaborative scheduling process, Agents iteratively repeat the auction and bid 20 times, and adjust the production schedule in order to generate the feasible daily production schedule, as shown in Figure 6-17. In Figure 6-17, the production scheduling cost in the system layer is plotted against the iterations.

Assume that the manufacturing profit coefficient of each processing stage in a lot is the same and the WIP inventory cost is 1. The manufacturing profit of the lot product from the first day to the third day is 7, 5 and 3, respectively. The manufacturing profit of these wafers from the fourth day to the seventh day is 3, 5 and 7, respectively. Figure 6-18 shows the daily scheduling cost of wafer fabrication. In Figure 6-18, the production scheduling cost of wafers is plotted against days. When both the release proportion and the WIP inventory cost are the same, the larger the manufacturing profit coefficient is, the lower the scheduling cost is. Therefore, the capacity distribution of machines is changed by adjusting the manufacturing profit coefficient of the processing stage in order to effectively control the scheduling cost.

Table 6-1 Configuration of SFL.

Machine Pool Number	Processing Type	Number of Machines per Machine Pool	Number of per Batch Processing (u/lot)	MTBF (f/h)	MTTR (f/h)
1	Ion implant	3	1	50	2
2	Ion implant	4	1	70	1
3	Diffusion	3	5	100	2
4	Diffusion	4	5	110	3
5	Etching	2	1	90	1
6	Etching	4	1	80	1
7	Etching	3	1	60	2
8	Etching	2	1	70	1
9	Lithography	4	1	90	1
10	Lithography	3	1	70	3
11	Lithography	2	1	100	1

Table 6-2 Lot process stages.

Stage No.	Number of Periods by Product A	Machine Pool No. of Product A	Processing Time for Product A in Key Machine (t/hour)	Number of periods by product B	Machine Pool No. of Product B	Processing Time for Product B in Key Machine(t/hour)	Number of Periods by Product C	Machine Pool no. of Product C.	Processing Time for Product C in Key Machine (t/hour)
1	1	8	1	1	10	1	1	10	1
2	1	7	1	1	8	1	1	8	1
3	1	10	1	4	5	1	2	5	1
4	1	9	1	1	10	1	1	8	1
5	5	0	1	1	1	1	1	0	1
6	3	0	1	1	6	1	4	3	6
7	4	2	4	1	10	1	1	8	1
8	1	8	1	1	9	1	2	5	1
9	1	5	1	6	0	1	2	2	3
10	1	3	3	1	0	1	1	8	1
11	1	0	1	1	4	1	2	5	3
12	2	10	1	1	8	1	1	0	1
13	1	0	1	2	1	1	2	3	6
14	4	4	2	3	2	3	1	8	1

(*Continued*)

Table 6-2 (Continued)

Stage No.	Number of Periods by Product A	Machine Pool No. of Product A	Processing Time for Product A in Key Machine (t/hour)	Number of periods by product B	Machine Pool No. of Product B	Processing Time for Product B in Key Machine(t/hour)	Number of Periods by Product C	Machine Pool no. of Product C.	Processing Time for Product C in Key Machine (t/hour)
15	1	6	1	1	10	1	2	9	1
16	2	1	1	1	8	1	2	4	1
17	3	4	2	2	5	2	3	0	1
18	1	9	1	1	3	1	1	8	1
19	1	5	1	2	5	1	3	6	1
20	1	8	1	1	9	1	2	1	1
21	3	3	3	1	1	1	1	8	1
22	1	5	1	2	5	1	3	7	1
23	4	2	4	1	9	1	2	2	3
24	1	9	1	3	0	1	2	5	1
25	3	4	2	3	2	3	1	0	1
26	2	0	1	2	10	1	1	9	1
27	1	8	1	2	7	1	2	4	1
28	1	7	1	/	/	/	1	1	1
29	2	3	3	/	/	/	/	/	/
30	1	2	1	/	/	/	/	/	/

Table 6-3 Results.

Date	Release Amount r/lot	Low Bound Cost	Feasible Cost	Dual Difference Rate w/%
1	30	−6964	−6393	8.2
2	36	−7535	−6687	11.26
3	24	−6797	−5960	12.31
4	30	−7012	−6064	13.52
5	36	−7948	−7326	7.82
6	24	−6881	−6251	9.16
7	45	−8081	−6916	14.42

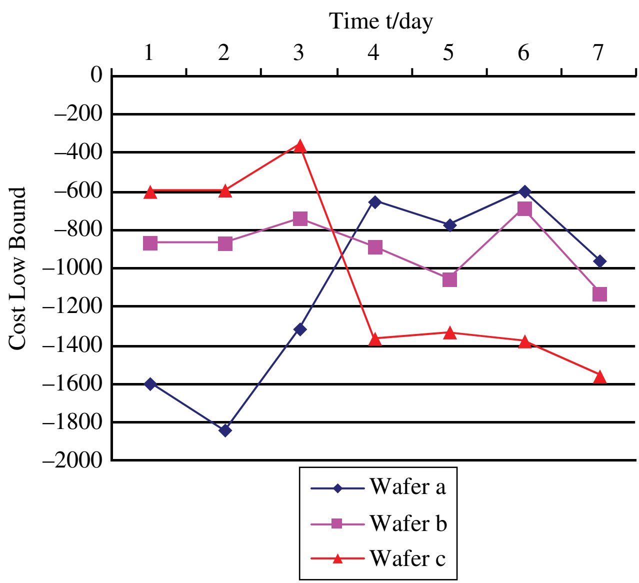

Figure 6-17 Result of Auction and Bid. Dual Difference Rate w=|(Feasible Cost—Low Bound Cost)/Low Bound Cost.

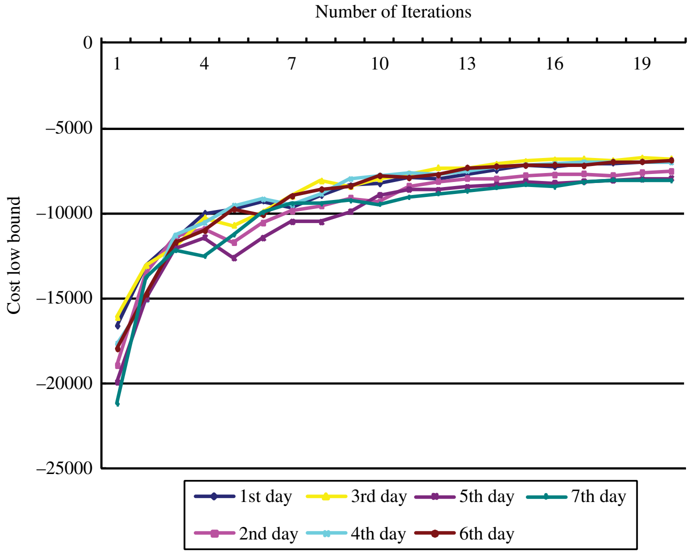

Figure 6-18 Scheduling cost of wafer fabrication (per day).

6.6 Adaptive Rescheduling in Re-Entrant Manufacturing Systems

In re-entrant manufacturing systems, the collaborative scheduling Agent is adopted to organize the task management Agent and the resource capacity management Agent to implement the adaptive rescheduling process in order to adjust the original production plan by analyzing the real-time information provided by the Multi-Agent production control system. In this section, a fuzzy neural network (FNN) based adaptive rescheduling method is proposed according to the information processing requirements of the collaborative scheduling Agent in the adaptive rescheduling process.

6.6.1 Problem Description

6.6.1.1 Disturbances

The re-entrant manufacturing system operates in uncertain dynamic system environments, where disturbances include machine breakdown, lot rework and rush orders. When disturbances happen, it is necessary to select the optimal rescheduling strategy on the basis of the current system status and the impact of disturbances to ensure the stability and effectiveness of the re-entrant manufacturing system. In order to quantitatively describe the impact of the disturbances on the system, it is noted that they are uniformly converted to the effects of disturbances on machine processing time. Hence, the influence of disturbances on the system scheduling process is presented as follows.

(1) Disturbance of machine breakdown

Assuming that M_{1j} is defective machine pool j; and m_{ji} is defective machine i in machine pool j with repair time t_{1ji} (hours). t_1 is the processing capacity reduction of the system caused by machine breakdown.

$$t_1 = \sum_{M_{1j} \in F_1} \sum_{m_{ji} \in M_{1j}} t_{1ji} \tag{6-32}$$

Where F_1 is the set of machine pools in which machine breakdown occurs.

(2) Disturbance of lot rework

Suppose that M_{2j} denotes the machine pool j to process rework tasks. R_{2j} is the rework task set processed in machine pool j. p_{jk} is rework part k processed in machine pool j with operation processing time t_{2jk}. t_2 is the processing capacity increasing demand of the system caused by rework tasks.

$$t_2 = \sum_{M_{2j} \in F_2} \sum_{p_{jk} \in R_{2j}} t_{2jk} \tag{6-33}$$

Where F_2 is the set of machine pools in which rework tasks are processed.

(3) Disturbance of rush order

M_{3j} denotes the machine pool j to process rush orders in the current scheduling period. R_{3j} is the task set including rush orders processed in machine pool j. p_{jk} is task k contained in the rush orders processed in machine pool j with operation processing time t_{3jk}. t_3 is the processing capacity increasing demand of the system caused by rush orders.

$$t_3 = \sum_{M_{3j} \in F_3} \sum_{p_{jk} \in R_{3j}} t_{3jk} \tag{6-34}$$

Where F_3 is the set of machine pools in which rush orders are processed.

Therefore, the effects of disturbances on the re-entrant manufacturing system are defined as follows.

$$T = t_1 + t_2 + t_3 \tag{6-35}$$

6.6.1.2 Performance of Rescheduling Process

The aim of the rescheduling strategy is to reduce the impact of disturbances on production schedules for re-entrant manufacturing systems, and to maintain better system performance after revising original production schedules. In this section, the fluctuation of both the objective function and the starting time is adopted to evaluate the robustness of the rescheduling strategy.

(1) Fluctuation of Objective Function Value

The fluctuation of the objective function indicates the relative changing rate of objective function values, which is measured from before adjusting to after adjusting

$$\alpha = \frac{|O_{new} - O_{old}|}{O_{old}} \tag{6-36}$$

Where O_{new} and O_{old} are the objective function values after adjusting and before adjusting, respectively.

(2) Fluctuation of Starting Time

The fluctuation of the starting time indicates the average changing rate of task starting time which is measured from before adjusting to after adjusting

$$\beta = \frac{\sum_{(i,s) \in S} |t'_{is} - t_{is}|}{\sum_{(i,s) \in S} q_{is} \cdot T_p} \tag{6-37}$$

Where t'_{is} and t_{is} are the starting times of product i processed in processing stage s after adjusting and before adjusting, respectively. S is the set of production schedules generated by system layer production scheduling. q_{is} is the quantity of product i processed in processing stage s. T_p is the length of the scheduling horizon.

(3) Robustness of rescheduling strategy

The robustness of a rescheduling strategy is defined as follows.

$$R = \frac{1}{\lambda \cdot \alpha + (1-\lambda) \cdot \beta} \quad (0 < \lambda < 1) \tag{6-38}$$

Where λ is the weighted coefficient.

6.6.2 Rescheduling Strategy

Considering the real-time status of the system and the influence of disturbances on the original production plan, it is necessary to introduce the hierarchical rescheduling strategy.

1) When the effects of disturbances on the re-entrant manufacturing system are large, the collaborative production scheduling strategy in the system layer is adopted. According to the current status of the system, the rescheduling strategy in the system layer is employed to revise production schedules in the system layer. The collaborative production scheduling algorithm in the machine layer for re-entrant manufacturing systems is used to implement real-time the production scheduling process for machines/machine pools on the basis of the results of the system layer rescheduling process.
2) When the effects of disturbances on the re-entrant manufacturing system are small, the rescheduling strategy in the machine layer is adopted. The starting times of lot operations must be adjusted while the operation sequences of the lots are kept from changing.
3) When the effects of disturbances on the re-entrant manufacturing system are between the above two, the rescheduling strategy in the machine pool layer is adopted. The collaborative production scheduling algorithm in the machine pool layer for re-entrant manufacturing systems is used to locally adjust production schedules for disturbed machine pools while production schedules of machine pools formulated by the collaborative production scheduling algorithm in the system layer are kept from changing.

6.6.3 FNN-Based Rescheduling

Due to the advantages of fuzzy logic theory in solving uncertain problems, expert knowledge such as fuzzy language is transformed as network weighted values, and the nodes in the neural network are given practical meanings in order to convert the rule selection problem to a network local-weighted estimation problem. Furthermore, the uncertain relationship between system status parameters, disturbance parameters and rescheduling strategies is identified and analyzed, and the nonlinear relationship between them is established by using a FNN. Then an appropriate rescheduling strategy can be selected based on current system running status when sudden changes in the actual manufacturing environment occurred.

6.6.3.1 Input Variables of FNN Model

The variables in the FNN input layer consist of state parameters and disturbance parameters T for re-entrant manufacturing systems. The state parameters include average queue length, stability of system and average relative loads.

1) w is the average queue length of machine pools affected by disturbances. It reflects the machine utilization of machine pool j.

$$w = \frac{\sum_{M_j \in (F1 \cup F2 \cup F3)} w_j}{R} \tag{6-39}$$

Where w_j is the queue length of machine pool M_j. R is the number of machine pools affected by disturbances.

2) β_c represents the scheduling stability of re-entrant manufacturing systems, which is defined as the deviation in predicted average starting time of a rescheduling strategy from the real starting time.

$$\beta_c = \frac{\sum_{\substack{(i,s) \in S \\ tc_{is} \le tc}} \left| tc'_{is} - tc_{is} \right|}{\sum_{\substack{(i,s) \in S \\ tc_{is} \le tc}} q_{is}} \tag{6-40}$$

Where tc'_{is} is the practical starting time of lot (processing stage s of product i). tc_{is} is the planned starting time of lot (processing stage s of product i) generated by system layer production schedule (middle time of scheduling time). t_c is the current time when disturbance happens.

3) Let η represent the average relative loads of machine pools affected by disturbances, which are measured from the current time to the end of the scheduling horizon.

$$\eta = \frac{\sum_{\substack{(i,s)\in S_d \\ t_e \geq t_{is} \geq t_c}} tp_{is}}{\sum_{M_j \in (F1 \cup F2 \cup F3)} n_j (t_e - t_c)} \tag{6-41}$$

Where tp_{is} is the operation processing time of processing stage s of product i. t_{is} is the starting time of processing stage s of product i. S_d is the set of machine pools affected by disturbances in the scheduling results generated by the system layer. n_j is the number of machines in machine pool M_j. In addition, one variable t_s represents the average slack, which is the schedule adjustment space given to machine pools affected by disturbances.

$$ts = \frac{\sum_{\substack{(i,s)\in S_d \\ t_e \geq t_{is} \geq t_c}} \left(t_{i(s+1)} - t_{is} - tp_{is}\right)}{\sum_{\substack{(i,s)\in S_d \\ t_e \geq t_{is} \geq t_c}} q_{is}} \tag{6-42}$$

6.6.3.2 FNN Structure

FNN consists of five layers, as shown in Figure 6-19. In FNN, the Mamdani-based fuzzy inference[48, 49] is used. Suppose the fuzzy rule R_i describes the relationship between input and output. Then

IF x_1 is A_{1i} and x_2 is A_{2i} and … and x_m is A_{mi} THEN y_1 is B_{1i} and y_2 is B_{2i} and … and y_m is B_{ki}, i=1,2, …,n.

Where n is the number of rules; m is the number of input variables; k is the number of output variables; A_{ji} is the value of fuzzy linguistic variable x_j; B_{ji} is the value of fuzzy linguistic variable y_j.

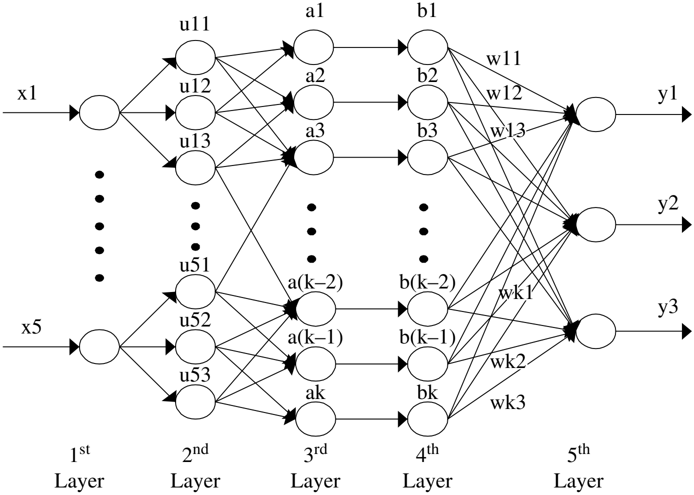

Figure 6-19 FNN Structure.

FNN is presented in detail as follows:

1) The first layer is the input layer. The input vector is X = [x_1, x_2, x_3, x_4, x_5] = [w, β_c, η, ts, T]. The node input-output function is:

$$f_i^{(1)} = x_i^{(0)} = x_i; x_i^{(1)} = g_i^{(1)} = f_i^{(1)}; i = 1,2,\cdots 5 \tag{6-43}$$

2) The second layer is the fuzzifier layer. The Gaussian membership function[50] is adopted.

$$u_{ij} = e^{-\frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}}, i = 1,2,\cdots,5, j = 1,2,\cdots,k_i \tag{6-44}$$

Where c_{ij} and σ_{ij} are the center and width variables, respectively. The node input-output function in this layer is defined as follows:

$$f_{ij}^{(2)} = -\frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}; x_{ij}^{(2)} = u_{ij} = g_{ij}^{(2)} = e^{f_{ij}^{(2)}} = e^{-\frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}};$$
$$i = 1,2,\cdots 5, j = 1,2,\cdots,k_i \tag{6-45}$$

3) The third layer is the rule layer. Each node represents a fuzzy rule that matches the front part of the fuzzy rule and calculates the adaptive value of the fuzzy rule.

$$a_j = \prod_{l=1}^{5} u_{li_l}\left(x_{li_l}\right), j=1,2,\cdots,n \tag{6-46}$$

The node input-output function in this layer is:

$$f_j^{(3)} = \prod_{l=1}^{5} x_{li_l}^{(2)} = \prod_{l=1}^{5} u_{li_l}\left(x_{li_l}\right); x_j^{(3)} = g_j^{(3)} = f_j^{(3)}; j=1,2,\cdots,n \tag{6-47}$$

4) The fourth layer is the normalized layer. Node numbers are equal to those in the third layer. In this layer, the adaptive values of rules are normalized.

$$b_j = \frac{a_j}{\sum_{i=1}^{n} a_i}, j=1,2,\cdots,n \tag{6-48}$$

The node input-output function in this layer is:

$$f_j^{(4)} = \frac{x_j^{(3)}}{\sum_{i=1}^{n} x_i^{(3)}} = \frac{a_j}{\sum_{i=1}^{n} a_i}; x_j^{(4)} = g_j^{(4)} = f_j^{(4)}; j=1,2,\cdots,n \tag{6-49}$$

5) The fifth layer is the output layer. It is used to defuzzify the output variables. Each node represents a rescheduling strategy. If a rescheduling strategy is selected, then the corresponding output is approximately 1, otherwise it is 0. The node input-output function in this layer is:

$$f_i^{(5)} = \sum_{j=1}^{n} w_{ij} x_j^{(4)} = \sum_{j=1}^{n} w_{ij} b_j; x_j^{(5)} = g_j^{(5)} = f_j^{(5)}; i=1,2,3 \tag{6-50}$$

Where w_{ij} is the connection weighted parameter.

6.6.3.3 Learning algorithm

The learning algorithm of FNN consists of structure learning and parameters learning. Structure learning comes first. Based on the statistics distribution of disturbances, a simulation model is developed for re-entrant manufacturing systems. When disturbances occur, a robust rescheduling strategy is selected based on effectiveness and stability. In this way, the mapping between system state parameters, disturbances, and rescheduling strategies is built up, and the learning sample is accumulated. Based on special rescheduling strategies, the samples are divided. Next, the fuzzy c-means cluster method is applied to classify the sample space. If there are n_i classes of samples, there are also n_i fuzzy rules. The total number of rules is $n = \sum n_i$. The center and radius of the cluster is shown as follows:

$$E = \frac{1}{2}\sum_{i=1}^{3}\left(y_i' - y_i\right)^2 \tag{6-51}$$

Where y_i' is the predicted output, and y_i is the actual output.

Parameter learning is based on the error back propagation algorithm. A supervised gradient descent algorithm is adopted to adjust FNN parameters w_{ij}, c_{ij} and σ_{ij}. The resulting parameter adjustment equations are:

$$w_{ij}(k+1) = w_{ij}(k) - \gamma\left(y_i' - y_i\right)b_j \tag{6-52}$$

$$c_{ij}(k+1) = c_{ij}(k) - \gamma \cdot \delta_{ij} \cdot \frac{2\left(x_i - c_{ij}\right)}{\delta_{ij}^2} \tag{6-53}$$

$$\sigma_{ij}(k+1) = \sigma_{ij}(k) - \gamma \cdot \delta_{ij} \cdot \frac{2\left(x_i - c_{ij}\right)^2}{\delta_{ij}^3} \tag{6-54}$$

Where $\gamma > 0$ is the learning rate and is positive.

$$\delta_{ij} = \sum_{k=1}^{n}\Big[\sum_{\gamma=1}^{3}\left(y_k' - y_k\right)w_{\gamma k}\frac{\sum_{\substack{i=1\\ i\neq k}}^{n} a_i}{\left(\sum_{i=1}^{n} a_i\right)^2}\Big]s_{ij}e^{-\frac{\left(x_i - c_{ij}\right)^2}{\sigma_{ij}^2}} \tag{6-55}$$

Where

$$s_{ij} = \begin{cases} \prod_{\substack{i=1\\ i\neq j}}^{5} u_{li_j}\left(x_{li_j}\right) & g_{ij}^2 = u_{ij} \text{ is the input of the jth rule node} \\ 0 & \text{others} \end{cases} \tag{6-56}$$

6.6.4 Case Study

The practical data collected from a 6-inch semiconductor fabrication line (SFL) in Shanghai is used to demonstrate the effectiveness of the proposed FNN-based rescheduling strategy. This SFL is composed of 11 key machine pools, which add up to 34 key machines with MTTF and MTTR parameters. There are three different types of lot products with 85 processing stages. In the simulation model for the production line, 150 sets of data are acquired by three rescheduling strategies, as shown in Tables 6-4; 90 sets of data are randomly selected as training samples, and another 60 sets of data are used as testing samples.

The proposed FNN-based rescheduling strategy for re-entrant manufacturing systems is run in Matlab 6.5. In this study, the fuzzy c-means cluster algorithm identifies 12 rules. During the FNN training stage, the learning rate γ was 0.01, and the initial value of w_{ij} was 1. The output of the FNN-based rescheduling strategy is shown in Table 6-5. It is obvious that the outputs of FNN-based rescheduling strategy are closer to the optimal rescheduling strategy. Figure 6-20 illustrates the mechanism of the network training process graphically. The output of FNN-based rescheduling strategy and its corresponding objectives are analyzed by using the linear regression analysis, as shown in Figure 6-21. It reveals that the optimal algorithm selected by the rescheduling strategy is an efficient algorithm.

6.7 Conclusion

In this chapter, a Multi-Agent-based hierarchical adaptive production scheduling method for re-entrant manufacturing systems has been introduced in detail. In the hierarchical production scheduling method, constraints related to machine preventive maintenance, the capacity of BPM and SPM, and production process are taken into consideration in the system layer. A Multi-Agent system combined with auction-based protocol in the system layer is used to optimize production schedules. Constraints related to the maximum batch size of BPM, machine breakdown, lot rework job and delivery due date are

Table 6-4 Samples of FNN training and testing.

Sample NO.	Average Queue Length of Disturbed Machine Pools x1/lot	Stability of Scheduling x2/h	Average Load of Disturbed Machine Pools x3	Average Slack Time of Disturbed Machine Pools x4/h	Disturbance x5/h	Rescheduling strategy		
						Machine Layer y1	Machine Pool Layer y2	System Layer y3
1	1	1.10	0.59	6.42	2.14	1	0	0
2	2	0.78	0.52	6.51	2.01	1	0	0
3	0	0.81	0.46	5.16	1.76	1	0	0
..	..	..	..	..	..	..	..	..
41	3	3.22	0.73	6.32	7.32	0	1	0
42	2	2.24	0.74	4.84	6.04	0	1	0
43	3	4.19	0.64	2.11	6.46	0	1	0
..	..	..	..	..	..	..	..	..
88	6	4.23	0.75	3.81	9.81	0	0	1
89	4	4.15	0.76	5.64	9.18	0	0	1
90	5	4.01	0.76	4.97	9.77	0	0	1

Table 6-5 Output of FNN-based rescheduling strategy.

Sample NO.	Average Queue Length of Disturbed Machine Pools x1/lot t	Stability of Scheduling x2/h	Average Load of Disturbed Machine Pools x3	Average Slack Time of Disturbed Machine Pools x4/h	Disturbance x5/h	Rescheduling strategy			FNN Selected Rescheduling strategy		
						Machine Layer y1	Machine Pool Layer y2	Production Line Layer y3	Machine Layer y1	Machine Pool Layer y2	Production Line Layer y3
1	0	1.07	0.48	5.81	1.21	1	0	0	0.9749	0.0375	0.0727
2	2	1.42	0.49	4.62	1.42	1	0	0	1.0744	−0.1905	0.1192
3	1	1.29	0.52	5.54	1.79	1	0	0	1.0051	−0.0021	0.0013
..	..	..	..	..	..	..	..	..	..	..	..
15	3	3.18	0.76	6.13	6.42	0	1	0	0.0261	0.8944	0.0902
16	3	3.34	0.76	5.73	5.76	0	1	0	0.0222	0.9557	0.0283
17	3	2.21	0.81	4.53	6.11	0	1	0	−0.0253	1.07	−0.0376
..	..	..	..	..	..	..	..	..	..	..	..
58	8	4.67	0.77	3.21	8.94	0	0	1	0.4339	−0.0253	0.6217
59	6	3.96	0.84	5.18	9.46	0	0	1	0.0179	0.2662	0.7766
60	7	3.75	0.72	4.76	9.71	0	0	1	−0.0715	−0.009	1.1457

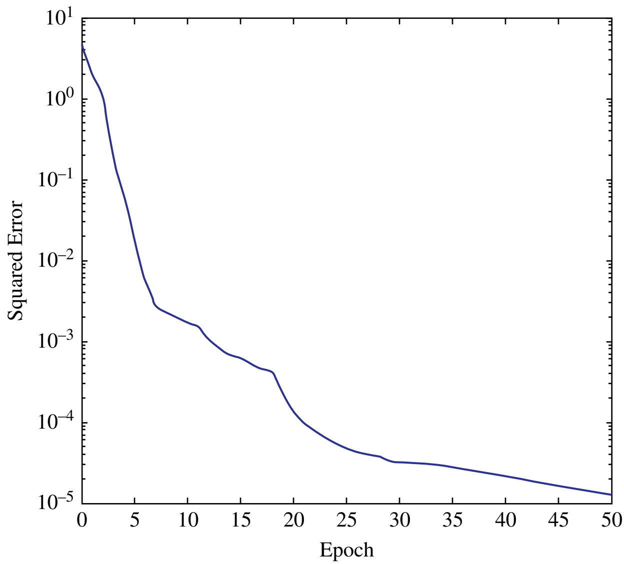

Figure 6-20 FNN learning process.

taken into consideration in the machine layer. A Multi-Agent GPGP-CN–based protocol in the machine pool layer is adopted to optimize production schedules in the machine pool. The objectives of the system layer are to maximize the profit of every processing stage and to minimize the WIP inventory cost, while the objectives in the machine pool layer are to improve machine utilization and to increase daily output. It is necessary to satisfy both the requirements in the system layer and those in the machine pool layer. In the adaptive rescheduling phase, considering the impact of constraints related to the maximum batch size of BPM, machine breakdown, lot rework, and delivery due date on the production schedule for machine pool, a FNN-based rescheduling method is presented in order to maximize the processing profit and to minimize the WIP inventory cost. It has provided an important tool to maintain the stability of re-entrant manufacturing systems in a dynamic environment.

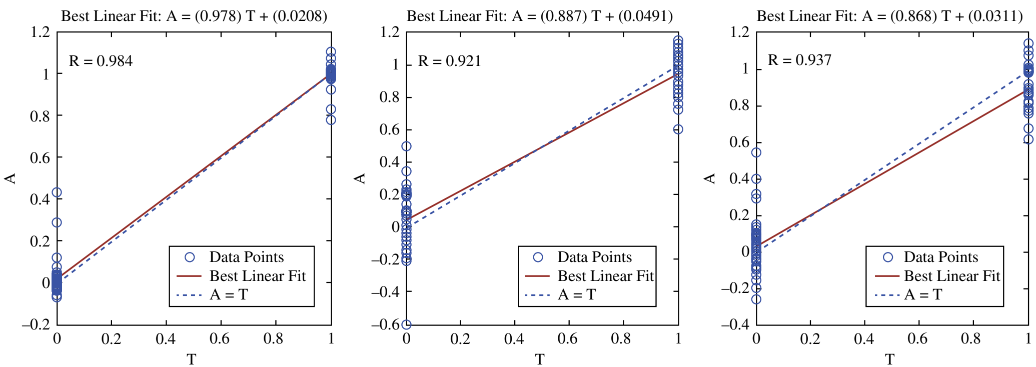

Figure 6-21 Linear regression analysis of FNN's output.

References

[1] Zhai Wenbin. Research on Intelligent Coordinated Control Technologies of Re-Entrant Manufacturing System. Ph.D. Thesis, Shanghai Jiaotong University, 2005.

[2] Hung, Y. F., and Leachman, R. C. A Production Planning Methodology for Semiconductor Manufacturing Based on Iterative Simulation and Linear Programming Calculations. *IEEE Transactions on Semiconductor Manufacturing*, 1996, 9(2): 257–269.

[3] Glassey, C. R., Shantikumar, J. G., and Seshadri, S. Linear Control Rules for Production Control of Semiconductor Fabs. *IEEE Transactions on Semiconductor Manufacturing*, 1998, 1(1): 36–46.

[4] Sung, C. S., Choung, Y. I., Kong, J. M., and Kim, Y. H. Minimizing Makespan on a Single Burn-In Oven with Job Families and Dynamic Job Arrivals. *Computers and Operations Research*, 2002, (29): 995–1007.

[5] Sarin, S., Shikalgar, S., and Shenai, V. Modeling and Analysis for the Reduction of Cycle Time at a Wafer Fabrication Facility. *Proceedings of the International Conference on Semiconductor Manufacturing Operational Modeling and Simulation*, 2001.

[6] Sung, C. S., and Choung, Y. I. Minimizing Makespan on a Single Burn-In Oven in Semiconductor Manufacturing. *European Journal of Operational Research*, 2000, (120), 559–574.

[7] Gupta, A. K., and Sivakumar, A. I. Multi-Objective Scheduling of Two-Job Families on a Single Machine: Omega. *International Journal Of Management Science*, 2005, 33(5): 399–405.

[8] Kumar, P. R. Re-entrant lines. *Queuing Systems*, 1993, 13(2): 87–110.

[9] Kumar, P. R. Scheduling Semiconductor Manufacturing Plants. *IEEE Control Systems*, 1994, 9(1): 33–40.

[10] Kumar, P. R. *Scheduling Manufacturing Systems of Re-Entrant Lines: Stochastic Modeling and Analysis of Manufacturing Systems*. New York: Springer-Verlag, 1994: 325–360.

[11] Liu, S. H., and Kumar, P. R. Distributed Scheduling Based on Due Date and Buffer Priorities. *IEEE Transactions on Automatic Control*, 1991, 36(12): 1406–1416.

[12] Lu, S. H., Ramaswamy, D., and Kumar, P. R. Efficient Scheduling Policies to Reduce Mean and Variance of Cycle-Time in Semiconductor Manufacturing Plants. *IEEE Transactions on Semiconductor Manufacturing*, 1994, 7(3): 374–385.

[13] Qiao Fei, Li Li, Wang Zuntong, et al. Improved Hybrid Rescheduling Strategy for Semiconductor Manufacturing. *Computer Integrated Manufacturing System*, 2007, 13(3): 558–562.

[14] Min, H., and Yuehwern, Y. Selection of Dispatching Rules on Multiple Dispatching Decision Points in Real-Time Scheduling of a Semiconductor Wafer Fabrication System. *International Journal of Production Research*, 2003, 41(16): 3921–3941.

[15] Elaine, R., and Kevin, K. *Artificial Intelligence*. New York: McGraw-Hill, 1991.

[16] Li Li, Fei Qiao, and Wu Qidi. Research on Rescheduling for Semiconductor Wafer Fabs. *China Mechanical Engineering*, 2006, 17(6): 616–621.

[17] Li Li, Qiao Fei, Xu Xiaohong, et al. Complete Modification Rescheduling Method for Semiconductor Wafer Fabs. *Computer Integrated Manufacturing System*, 2006, 12(7): 1022–1027.

[18] Savell, D., Perez, R., and Koh, S. Scheduling Semiconductor Wafer Production: An Expert System Implementation. *IEEE Expert*, 1989, 4(3): 9–15.

[19] Liu, A. Shr, M. D., and Cheng, Y. A Fuzzy Scheduling System for Dedicated Machine Constraint. *Proceedings of the 9th Joint Conference on Information Sciences*, 2006: 72.

[20] Fargher, H. E., Kilgore, M. A., Kine, P. J., et al. Planner and Scheduler for Semiconductor Manufacturing. *IEEE Transactions on Semiconductor Manufacturing*, 1994, 7(4): 117–126.

[21] Huang, H., and Chen, T. A New Approach to On-Line Rescheduling for a Semiconductor Foundry Fab. *IEEE International Conference on Systems*, Man and Cybernetics, 2007: 4727–4732.

[22] Ding Yongsheng. *Computation Intelligence – Theory, Technology and Application*. Beijing: Science Press, 2004.

[23] Qu, P., and Mason, S. J. Using Tabu Search on the Single Machine Multi-Orders per Job Scheduling Problem. *IIE Annual Conference and Exhibition*, 2004: 1831–1835.

[24] Geiger, C., Kempf, K. G., and Uzsoy, R. A Tabu Search Approach to Scheduling an Automated Wet Etch Station. *Journal of Manufacturing Systems*, 1997,16(2): 102–116.

[25] Bhushan, S., and Karimi, I. A. Heuristic Algorithms for Scheduling an Automated Wet-Etch Station. *Computers and Chemical Engineering*, 2004, 28(3): 363–379.

[26] Toba, H. Segment-Based Approach for Real-Time Reactive Rescheduling for Automatic Manufacturing Control. *IEEE Transactions on Semiconductor Manufacturing*, 2000, 13(3): 264–272.

[27] Bussmann, S., and Schild, K. Self-Organizing Manufacturing Control: An Industrial Application of Agent Technology. *Proceedings of Fourth International Conference on Multi-Agent Systems*, 2000: 87–94.

[28] Massotte, P., and Bataille, R. Future Production System: Influence of Self-Organization on Approaches to Quality Engineering. *International Journal of Production Economics*, 2000, 64: 359–377.

[29] Kubota, N., and Fukuda, T. Structured Intelligence for Self-Organizing Manufacturing System. *Journal of Intelligent Manufacturing*, 1999, 10: 121–133.

[30] Albert, D. B. A Survey of Factory Control Algorithms That Can Be Implemented in a Multi-Agent Hierarchy: Dispatching, Scheduling, and Pull. *J. of Manufacturing Systems*, 1998, 17(4): 297–320.

[31] Huguet, P., and Grabot, B. A Conceptual Framework for Shopfloor Production Activity Control. *International Journal of Computer Integrated Manufacturing*, 1995, 8(5): 357–369.

[32] Jones, A., and Saleh, A. A Multi-Level/Multi-Layer Architecture for Intelligent Shopfloor Control. *International Journal of Computer Integrated Manufacturing*, 1990, 3(1): 60–70.

[33] Lyons, G. J., Duggan, J., and Bowden, R. Project 477: Pilot Implementation of a Production Activity Control (PAC) System in an Electrical Assembly Environment. *International Journal of Computer Integrated Manufacturing*, 1990, 3(3): 196–206.

[34] Pluym, V. Knowledge-Based Decision Making for Job-Shop Scheduling. *International Journal of Integrated Manufacturing*. 1990, 3(5): 354–363.

[35] Uzsoy, R., Lee, C-Y., and Martin-Vega, L. A. A Review of Production Planning and Scheduling Models in the Semiconductor Industry. Part I: System Characteristics, Performance Evaluation and Production Planning, *IIE Trans*, 1992, 24(4): 47–60.
[36] Hynynen, J. A Framework for Coordination in Distributed Production Management. Helsinki University of Technology, Dissertation, 1988.
[37] Zhang, W., and Mao, Z. J. Synthesis of an Event Based Supervisor for Deadlock Avoidance in Semiconductor Manufacturing Systems. *Proceedings of the 2004 American Control Conference*, 2004: 1417–1421.
[38] Ting-Kai, H., and Shi-Chung, C. Design of a Lagrangian Relaxation-Based Hierarchical Production Scheduling Environment for Semiconductor Wafer Fabrication. *IEEE Transactions on Robotics and Automation*, 2003, 19(4): 566–579.
[39] Wang Zhongjie and Wu Qidi. Research on Control and Scheduling of Semiconductor Production Line. *Computer Integrated Manufacturing System*, 2002, 8(8): 607–611.
[40] Odrey, N. G., and Mejia, G. A. Re-Configurable Multi-Agent System Architecture for Error Recovery in Production Systems. *Robotics and Computer Integrated Manufacturing*, 2003, 19: 35–43.
[41] Karl, G. Scheduler/Dispatcher User Requirements. *International SEMATECH*, 2000.
[42] Edgar, T. F. Automatic Control in Microelectronics Manufacturing: Practices, Challenges, and Possibilities. *Automatics*, 2000, 36(11): 1567–1603.
[43] Zhu Jianying. New Developments on System Models, Modeling Methods and Key Technologies for Modern Manufacturing. *Chinese Journal of Mechanical Engineering*, 2000, 36(8): 1–5.
[44] Giddings, R. A. Pervasive Integrated Material Handling in 300mm Semiconductor Manufacturing. *Proceedings 2004 IEEE/SEMI-Advanced Semiconductor Manufacturing Conference and Workshop*, 2004: 274–278.
[45] Dietrich, J. M. Life Cycle Process Management for Environmentally Sound and Cost Effective Semiconductor Manufacturing. *Proceedings of the 2004 IEEE International Symposium on Electronics and the Environment*, 2004: 168–172.

[46] Gonia, M. Using Manufacturing Execution Systems (MES) to Track Complex Manufacturing Processes. *Proceedings of the IEEE/CPMT International Electronics Manufacturing Technology (IEMT) Symposium,* 2004: 171–173.

[47] Chen, T.-R., Chang, T.-S. and Chen, C.-W. Scheduling for IC Sort and Test with Preemptiveness via Lagrangian Relaxation. *IEEE Transactions on Systems, Man and Cybernetics*, 1995, 25(8): 1249–1256.

[48] Liu, P. Mamdani Fuzzy System: Universal Approximator to Class of Random Processes. *IEEE Transactions on Fuzzy Systems*. 2002, 10(6): 756–766.

[49] Duan, J., and Chung, F. Mamdani Type Multistage Fuzzy Neural Network Model. *IEEE International Conference on Fuzzy Systems.* 1998, 2: 1253–1257.

[50] Kumar, S. *Neural Networks*. Beijing:Tsinghua University Press, 2006.

7

Multi-Agent-Based Production Control

7.1 Introduction

The results of production scheduling in Job Shop manufacturing systems or re-entrant manufacturing systems are delegated to the shop floor in the form of task orders. The aim of the production layer in manufacturing systems is to ensure accurate running of the manufacturing process, to coordinate all the resources (e.g., machines, materials, humans and so on) and to track the manufacturing process. The manufacturing process is tracked in order to obtain, analyze and feed back the information related to the raw materials, machines, operators and processes. As pointed out in Chapter 3, this production control process is defined as a typical pull production control process. Although the traditional black box method is adopted by most companies, it is hard to obtain real-time information, which leads to insufficient accurate real-time production data to support the planning and scheduling layer so as to promptly and effectively respond to the complicated manufacturing environment. In order to overcome the disadvantage of the traditional production control manner, a technology called Agent with mobility and proactive properties has been introduced as an alternative to obtain information related to raw materials, machines, operators and processes in the production control procedure in order to implement feedback and analysis functions.

7.2 Multi-Agent Production Control System

7.2.1 Requirements of Production Control Process

Production control process can be described as follows:

- The production control process exists in the bottom layer of the production planning and control system;
- The production control process is an important function of production management;
- The production control process provides an efficient means to achieve objectives of the production planning and production scheduling process;
- The production control process is a collection of activities to achieve the predefined production planning target, by which supervision and inspection of the actual production process, deviation detection, adjustment and correction are carried out in accordance with its corresponding arrangements and standards of production planning and scheduling process.

The pull production control system is introduced in this book. The production control system is required to formulate real-time feedback of production progress, to track the production process, to analyze the production data and to provide warning signs of abnormal states by acquiring real-time data about production procedures. The specific requirements of the pull production control process in manufacturing systems are summarized as follows:

1) Production Task Allocation
   In this phase, the production control system is adopted to receive production plans released by the Multi-Agent production scheduling system, then allocates task orders to material managers and machine managers in order to ensure that material and resources are well prepared in the required time, and are ready to execute the production process.
2) Production Process Management
   During the implementation process of production tasks, it is necessary for the production control system to obtain statistics of all kinds of progress data, production process quality

data, machine status data and production quantities, and thus to analyze and visualize these statistics.

3) Data Acquisition
   In this phase, the production control system is used to provide real-time accurate production data for the production control process to ensure the traceability of the production process.

7.2.2 The Architecture of a Multi-Agent Production Control System

Due to the initiative, mobility and other features of Agents, an Agent-based production control system is proposed in this book. According to the requirements of the pull production control process, a Multi-Agent production control system is developed, as shown in Figure 7-1.

Agents in the Multi-Agent production control system are presented in detail as follows:

1) Collaborative Task Management Agent
   The collaborative task management Agent is used to receive the production plan released by the Multi-Agent production planning system, and to convert it into task orders. The collaborative task management Agent is adopted to maintain the information of task orders, which includes Material No., quantity, the starting time, and the delivery due date. The collaborative task management Agent is employed to track the execution process of task orders, and to feed back the task execution status information to the Multi-Agent production planning system.
2) Machine Management Agent
   The machine management Agent represents the fundamental information of processing machines or assembling machines, and maintaining machines, which includes functions, attributes, ability levels, membership units, history records, depreciation and so on. When the machine has received a task order, the machine management Agent is used to record the information of the task required to be processed in the machine. When the status of the machine changes, the machine management Agent is adopted to interact with the OPC MAS in order to

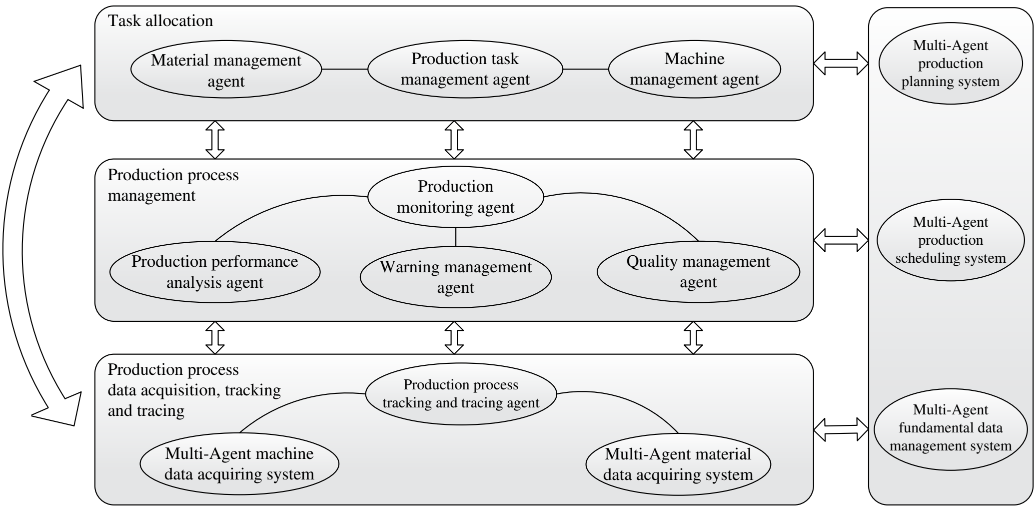

Figure 7-1 The architecture of Multi-Agent production control system.

obtain the real-time running information on machines, and to automatically update the data.

3) Material Management Agent
   The material management Agent is used to record the information related to material inventory and work-in-process inventory. When the materials are produced, the material management Agent is adopted to save the real-time status information of each product. The Multi-Agent RFID system is used to collect material data, and to interact with the material management Agent so as to obtain the real-time information of the material management Agent.
4) Production Monitoring Agent
   The production monitoring Agent is used to visualize the data obtained by the warning management Agent, the performance analysis Agent, the quality management Agent and the production process tracking and tracing Agent. The statistical analysis information based on the above data is also visualized by the production monitoring Agent in order to monitor the production process on line, along with the product quality and the status of machines.
5) Warning Management Agent
   The warning management Agent is used to build up a production-incidents library. The warning management Agent is adopted to generate warnings by analyzing the real-time information of the production process, and to track warning events and its solving process so as to ensure that abnormal events in the shop floor are detected and solved in time.
6) Performance Analysis Agent
   The performance analysis Agent is used to carry out statistical analysis for the production process, which includes the amount of raw material input, throughput, scrap rate, repair rate, machine utilization, and so on.
7) Quality Management Agent
   The quality management Agent is adopted to develop quality standards for the manufacturing process. The quality management Agent is used to acquire the quality data of the production process so as to analyze the quality, and to record the analysis results.

8) Production Process Tracking and Tracing Agent
The production process tracking and tracing Agent is adopted to generalize the material processing information and machine information incurred in the manufacturing process according to the material No. so as to support the product information inquiry and product backtracking to realize the historical information inquiry, and to obtain all the processing parameters corresponding to this product.
9) Multi-Agent Material Data Acquiring System
The Multi-Agent material data acquiring system is employed to implement real-time material data acquisition for manufacturing systems, which will be presented in detail in Chapter 8.
10) Multi-Agent Machine Data Acquiring System
The Multi-Agent machine data acquiring system is adopted to implement real-time machine data acquisition for manufacturing systems, which will be presented in detail in Chapter 9.

7.2.3 The Running Model for Multi-Agent Production Control Systems

A Multi-Agent production control system is a running process based on several Agents, which consists of three functions: production task allocation, production process management and data acquisition. In the production task allocation phase, the collaborative task management Agent is used to receive the production plan release by the Multi-Agent production scheduling system to generate task orders, and to send task orders to the machine management Agent. In addition, the collaborative task management Agent is adopted to send material delivery requests to the material management Agent. The material management Agent is employed to distribute material, while the machine management Agent is used to manage the implementation information of the manufacturing process, and to feed back the task execution information. Figure 7-2 illustrates the mechanism of production task allocation in Multi-Agent production control systems graphically.

The production process is managed jointly by the production monitoring Agent, the warning management Agent, the

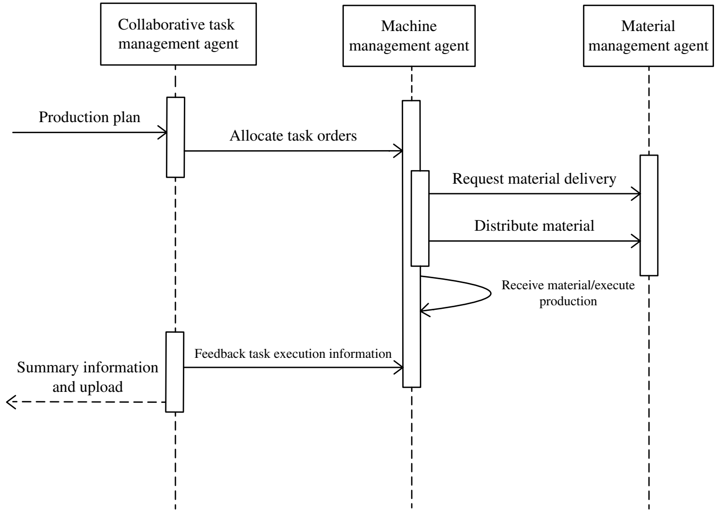

Figure 7-2 Production task allocation in Multi-Agent production control systems.

performance analysis Agent, and the quality management Agent in order to analyze and visualize the data of the production process. First, each Agent is used to analyze and transform the real-time data of the production process provided by the production process tracking and tracing Agent. Next, the production monitoring Agent is adopted to visualize the data. The collaborative procedure of the production process management in Multi-Agent production control systems is illustrated in Figure 7-3.

The production process tracking and tracing Agent is used to track the production data, and to provide real-time accurate production data to the Multi-Agent production control system. Data acquisition is the basis of a Multi-Agent production control system. The data acquisition process consists of a machine data acquisition process and a material data acquisition process, which are completed by a Multi-Agent machine data acquiring system and a Multi-Agent material data acquiring system, respectively. Their implementation processes will be presented in detail in Chapter 8 and Chapter 9.

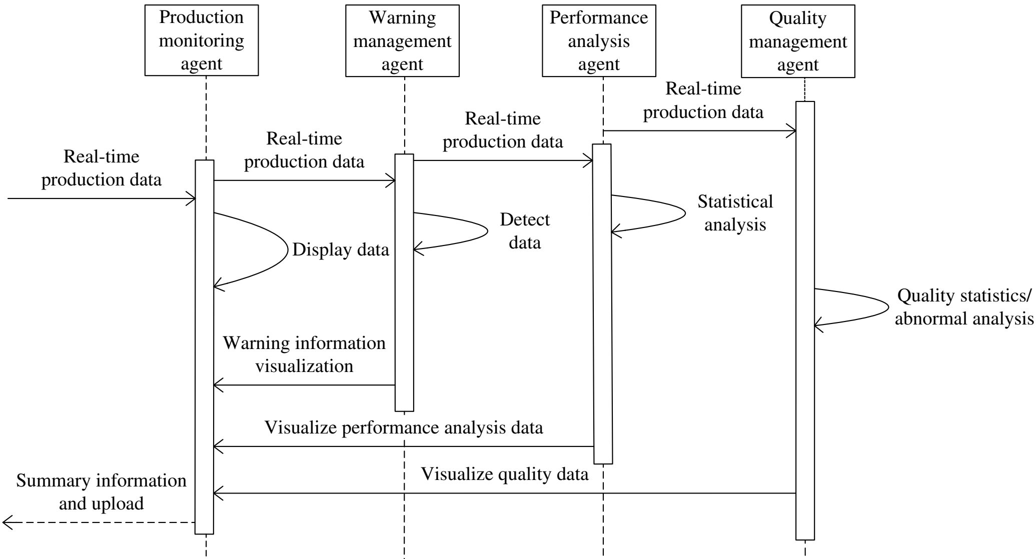

Figure 7-3 Production process management in Multi-Agent Production Control Systems.

7.3 Agents in Multi-Agent Production Control Systems

7.3.1 Collaborative Task Management Agent

The collaborative task management Agent is used to receive the production plan and to send clear instructions to operators on the shop floor. These instructions should clearly specify the information of each task to manufacture a part by using one machine in a period. The collaborative task management Agent is also adopted to record production indexes including capacity utilization, material shortage, efficiency of progress and so on. Whether the material is ready will be investigated before the product is processed on the production line, the implementation of task orders will be monitored after the product is processed on the production line. The behavior model of the collaborative task management Agent is illustrated in Figure 7-4.

1) Receive production plan and generate task orders
   In this phase, the collaborative task management Agent is adopted to interact with the Multi-Agent production scheduling system to obtain the production plan for manufacturing systems, and thus to generate task orders that clearly point

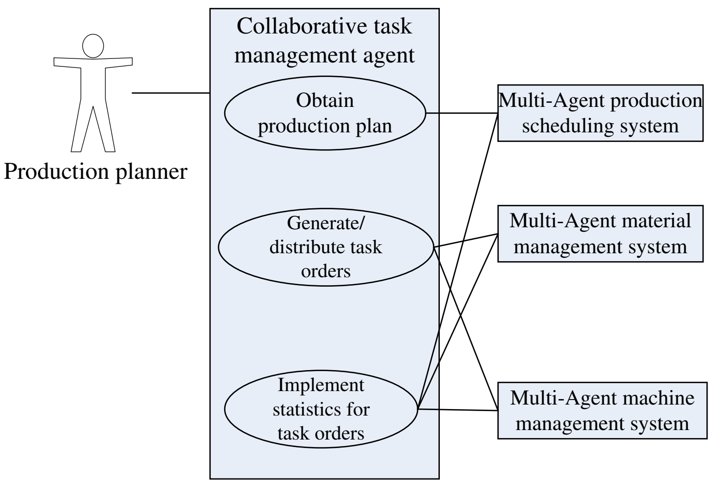

Figure 7-4 The use case diagram of a collaborative task management Agent.

out any product that is produced in any machine with any material in any period.

2) Distribute task orders

In this phase, the collaborative task management Agent is used to send task orders to the machine management Agent and the material management Agent to check the preparation status of material and production capacity in order to ensure that material and production capacity are ready.

3) Implement statistics for task orders

In this phase, according to scan results of the production material obtained by using RFID, the collaborative task management Agent is adopted to record the implementation status of task orders in order to implement statistics for this implementation status including production trends, completion rate of task orders, and so on.

Since the procedures of both obtaining the production plan and distributing task orders are a fixed process, the statistical results of implementing task orders are obtained by using many methods on the basis of parameters provided by the sensors, the collaborative task management Agent is regarded as a typical reactive Agent, and its mechanism is presented in Figure 7-5. In Figure 7-5, the information processing unit consists of the decomposition rules of the production plan to generate task orders and statistical rules of task orders implementation, and so on.

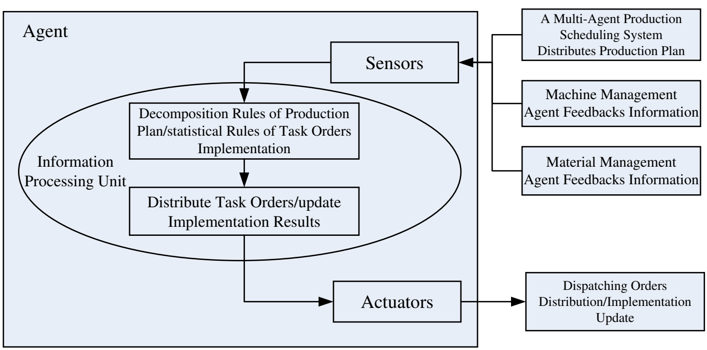

Figure 7-5 The mechanism of the collaborative task management Agent.

7.3.2 Machine Management Agent

The machine management Agent is adopted to manage processing or assembling machines in manufacturing systems. Its main functions are presented as follows:

1) Maintain machine fundamental information
   The machine fundamental information contains functions, properties, capacity levels, membership units, history records, depreciation of machines and so on.
2) Receive task orders
   When a task order is allocated to a machine, the machine management Agent is used to record the information of the task required to be processed in this machine.
3) Update machine status
   When the machine status changes, the machine management Agent is used to interact with the Multi-Agent OPC system in order to obtain the real-time running information of machines, and to automatically update the data.

In terms of the above functional analysis, the behavior model of the machine management Agent is developed, as shown in Figure 7-6.

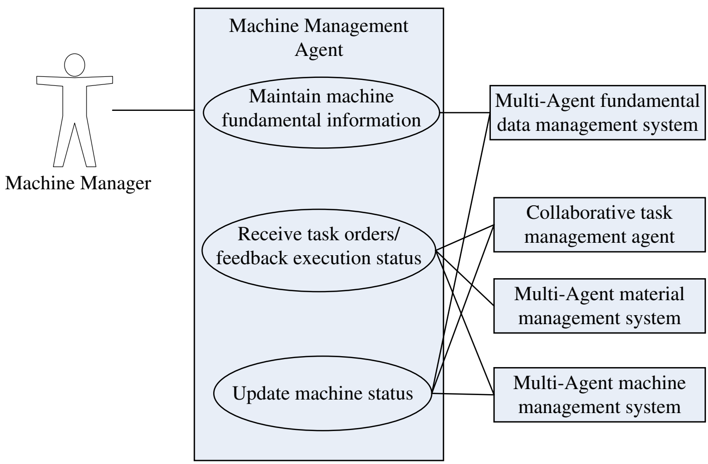

Figure 7-6 The use case diagram of a machine management Agent.

In terms of simple rules, the machine management Agent only needs to maintain machine basic information, to receive task orders/submit feedbacks, to update machine status and so on. Therefore, the machine management Agent is a reactive Agent.

7.3.3 Material Management Agent

The material management Agent is used to record information related to material inventory and work-in-process inventory. When the product is processed, the material management Agent is adopted to record the real-time status information of each work-in-process. The material management Agent is employed to interact with the Multi-Agent material data acquisition system in order to provide real-time material information. The behavior model of material management Agent is shown in Figure 7-7.

The material management Agent is similar to the machine management Agent, which is responsible for real-time updating and recording of the information related to warehouse inventories and the quantity of work-in-process. The material management Agent belongs to the reactive Agents.

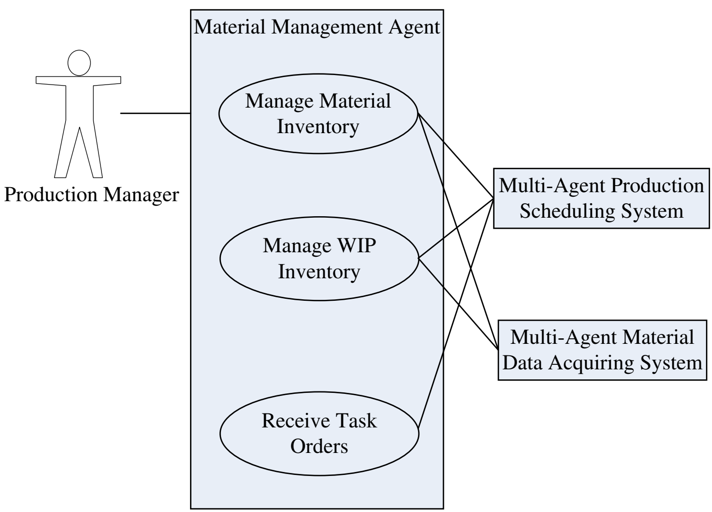

Figure 7-7 The use case diagram of a material management Agent.

7.3.4 Production Monitoring Agent

With the development of more complex technologies, the shorter cycles for various products – from their introduction to the stage of stable production – the more challenges are incurred in introducing new products. Similarly, the more complex product functions become, the more difficulties increase to design a new product. Hence, a manual or semi-automated monitoring system is no longer functional for flexible product manufacturing processes. At the same time, the complexity and precision of manufacturing systems are becoming more sophisticated; while the size of work in process is shrinking, there is need to improve the management level in manufacturing processes. The production monitoring Agent is used to visualize the data obtained by the warning management Agent, the performance analysis Agent, the quality management Agent and the production process tracking and tracing Agent. The statistical analysis information based on the above data is also visualized by the production monitoring Agent in order to monitor the production process on line, the product quality and the status of machines. The behavior model of production monitoring Agent is shown in Figure 7-8.

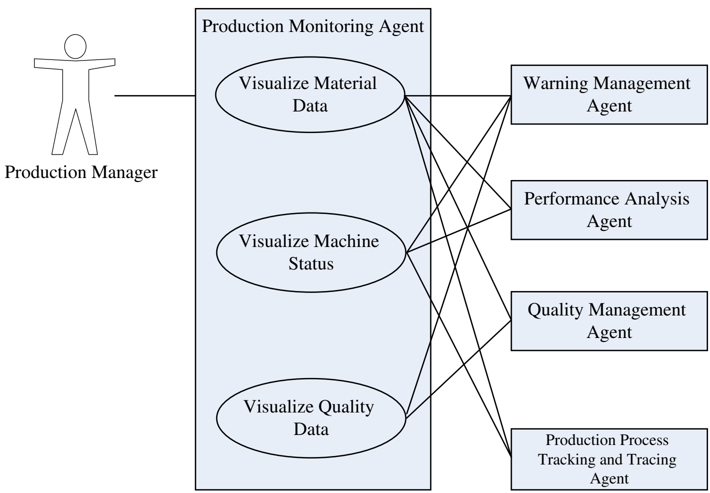

Figure 7-8 The use case diagram of a production monitoring Agent.

The production monitoring Agent is adopted to obtain the data from each production management Agent, and to display the data, which makes it a reactive Agent.

7.3.5 Warning Management Agent

When machine breakdown or scrap occurs in production processes, the warning management Agent acquires the real-time data. When the data exceeds the allowable range, the warning management Agent is adopted to provide alarms prompts to guide production managers to make decisions. The behavior model of the warning management Agent is shown in Figure 7-9.

The warning management Agent is used to obtain data from other related production management Agents and to display data with statistical analysis, which belongs to reactive Agents. The steps for implementing the warning management Agent are outlined clearly in Figure 7-10.

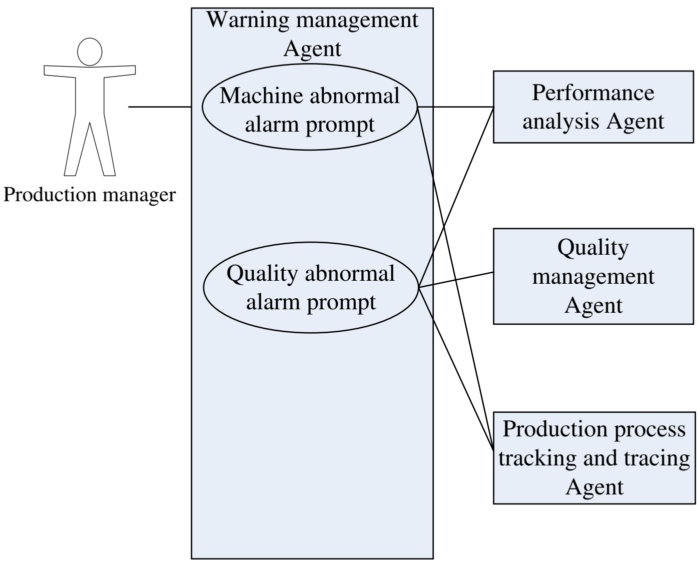

Figure 7-9 The use case diagram of a warning management Agent.

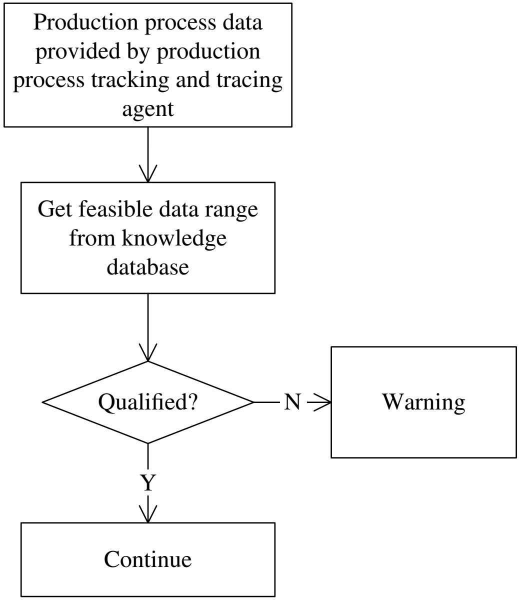

Figure 7-10 Workflow of abnormal warnings.

7.3.6 Performance Analysis Agent

The performance analysis Agent is used to carry out statistical analysis for the production process, which includes the amount of raw material input, throughput, scrap rate, repair rate, machine utilization and so on. The behavior model of the performance analysis Agent is shown in Figure 7-11.

Statistical results of the machine operating and material processing status can be acquired in the production process or at the end of the production process. The data related to the machine operation processing time, the machine operational efficiency, the total processing capacity, performance and product quality index are calculated and displayed by using the performance analysis Agent, respectively. The statistical workflow of the production process data is shown in Figure 7-12. In this book, the performance analysis Agent is developed based on given methods to implement statistical analysis without involving complex business logic, which is regarded as a reactive Agent.

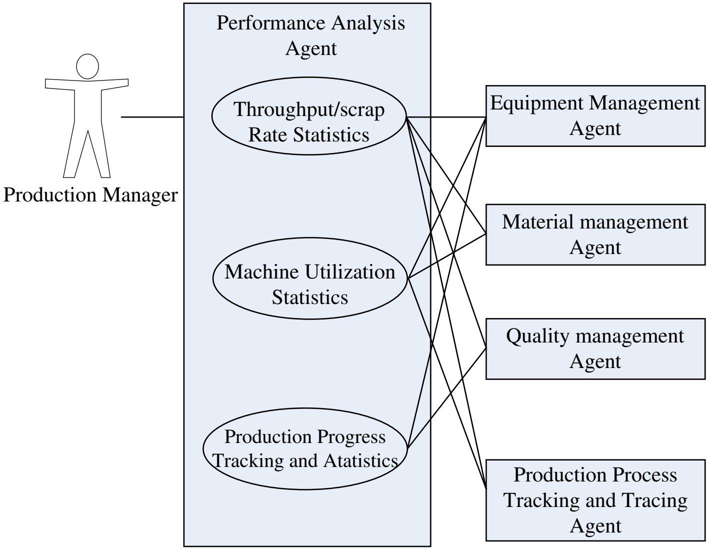

Figure 7-11 The use case diagram of a performance analysis Agent.

7.3.7 Quality Management Agent

Quality management is a process to continually improve key quality indexes based on the quantitative analysis in order to achieve excellent standards and to significantly improve enterprise performance and business performance by acquiring critical data of products and integrating a variety of knowledge and methods. In general, quality management consists of two aspects: quality analysis and quality control.[1] Quality analysis tracks quality factors in production processes and provides a variety of information queries and statistical analyses according to the management demands of administrations. Quality control provides improvements based on results of quality analysis. In terms of the above function analysis, the behavior model of the quality management Agent is developed in Figure 7-13.

The quality management Agent is a thinking Agent; its mechanism is shown in Figure 7-14. In Figure 7-14, the database is used to store the data acquired by material and quality testing devices, quality information of products, and results of data

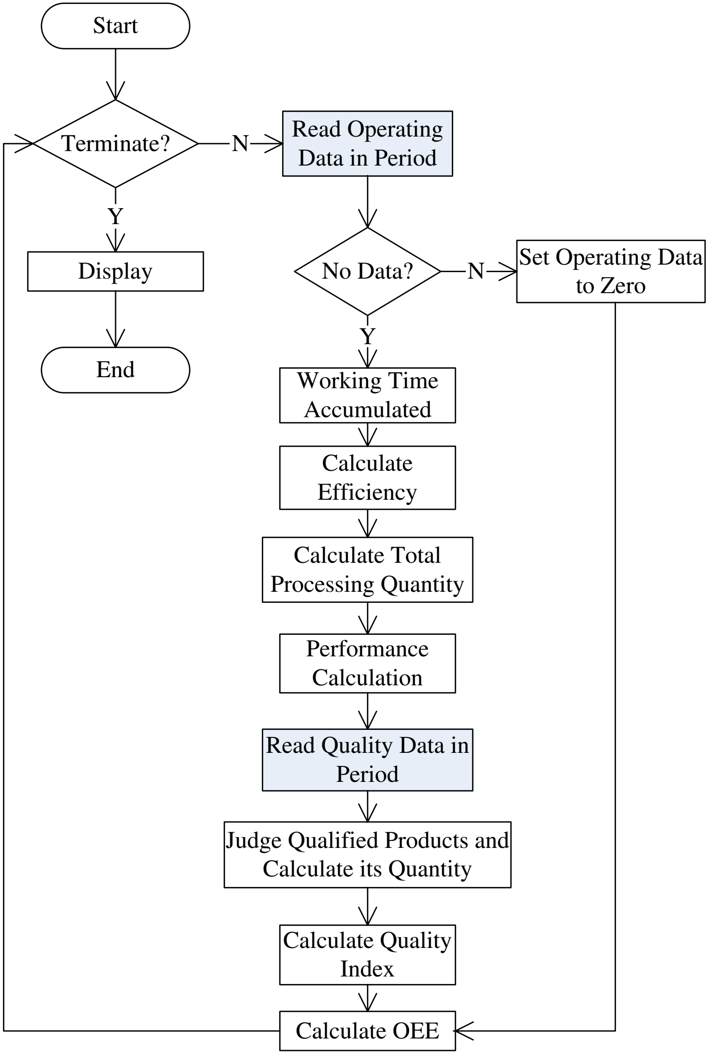

Figure 7-12 Statistical workflow of production process data.

analysis. The control logic is adopted to design in accordance with the functional requirements of the quality management Agent, which consists of the quality analysis algorithm and the quality control algorithm, and so on.

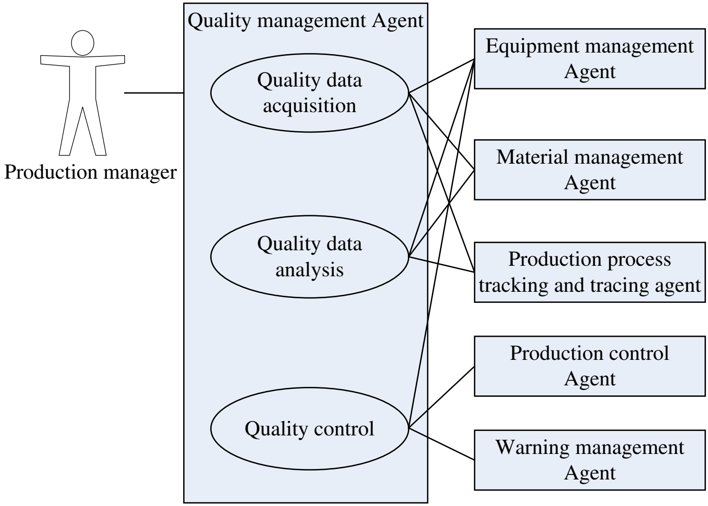

Figure 7-13 The Use case diagram of a quality management Agent.

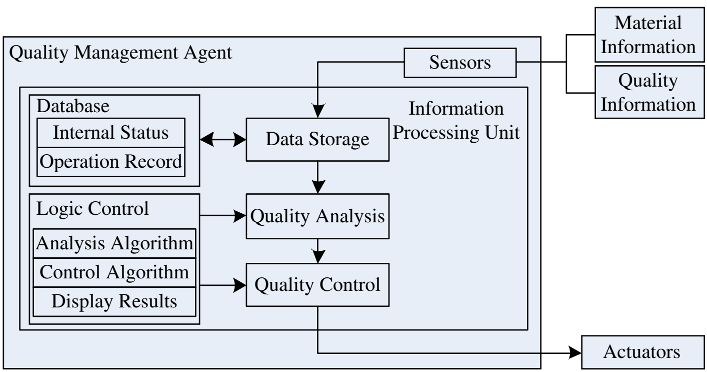

Figure 7-14 The mechanism of the quality management Agent.

7.3.8 Production Process Tracking and Tracing Agent

The purpose of production process tracking is to achieve traceability. However,there has been no rigorous definition of "traceability" until now. As pointed out in ISO8402 (Quality management and quality assurance-terms), traceability is the capability to track the history, the application and judgments of an object by recording information.[2] The definition of traceability made by APICS[3] in logistics view has two meanings: the first is to determine the location of goods in transportation; the second is to record and track parts, processes and material by lot numbers or serial numbers.

Traceability[4] has two meanings: tracking and backtracking. Tracking is a data recording process that follows the same direction as the production process, which records real-time information from raw material to products leaving the warehouse. Backtracking is a data recording process that moves in the opposite direction from the production process, which retrieves and displays the historical data process from products or semi-finished products. Production process tracking processes the status and events of dynamic production processes based on real-time data. Backtracking based on historical information has two meanings: the historical information inquiry of production processes and the backtracking of finished products. The former one runs inside the shop floor, which provides inquiries to the information related to workers, machines and material in production processes. The latter one runs outside the shop floor or even outside the enterprise, providing inquiries to the information related to batch number, quality and raw material. In particular, recalling products that the industry attaches great importance to is an important part of backtracking.

Based on the above definition, two main functions of production process tracking are summarized as follows:

1) Correlate and record production data
   The system is required to correlate and integrate the relevant data (including operator, machines, quality and material) acquired in production processes, and to record the information in the production processing process in detail, so as to develop the product information database to achieve real-time tracking of production processes.
2) Inquire production process information and backtrack product information
   In the production process, various statistical data of production process information is acquired so as to support optimal control of the production process. Meanwhile, when products are in the market, the backtracking function is enabled

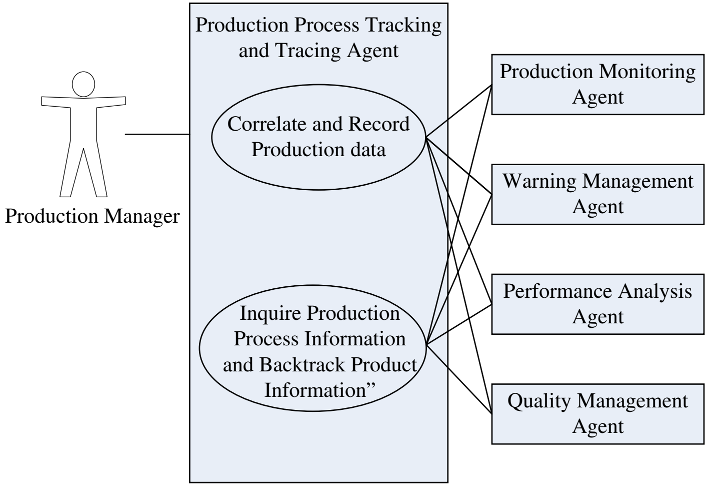

Figure 7-15 The use case diagram of a production process tracking and tracing Agent.

for customers to backtrack production process information for each product. The production process information includes raw material, machines, operators, and processes.

The behavior model of production process tracking and tracing Agent is shown in Figure 7-15.

In terms of the tracking functional analysis of the production process, the production process tracking and tracing Agent is adopted to analyze and process the real-time production process data acquired by data acquiring system, and then to provide the data for customers to back track the production process. Therefore, the production process tracking and tracing Agent is regarded as a thinking Agent, which should be given a special internal structure and control logic design as shown in Figure 7-16.

1) Database

The database is adopted to store the data acquired from the material and machines in the physical layer by using the production process tracking and tracing Agent, and to record various operating status information about machines and material. Meanwhile, the database is also used to save results from the analysis of production process data.

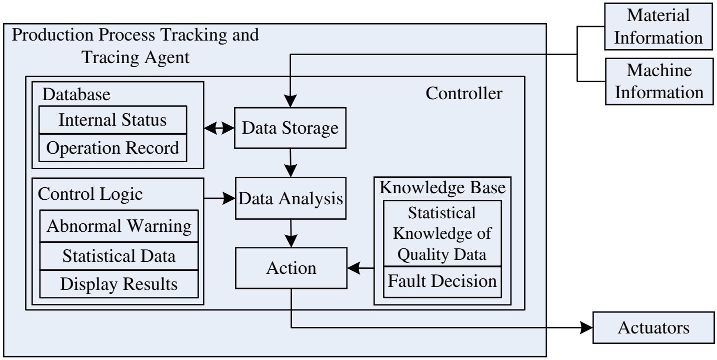

Figure 7-16 The mechanism of a production process tracking and tracing Agent.

2) Control logic
   According to the function requirements of the production process tracking and tracing Agent, the thinking Agent needs to design the logic process, which includes data correlate logic, data inquiry logic and so forth.
3) Knowledge base
   The knowledge base is used to provide knowledge support for the decision-making process of control logic, and to provide statistical knowledge of quality data for statistical logic.

7.4 Technologies and Methods for Multi-Agent Production Control Systems

7.4.1 XML-Based Production Monitoring

The main functions of the production monitoring Agent are to obtain and display real-time data, which visualize the production process. Its main purpose is to acquire and display, in real time, the information related to the production process. In particular, the status of a machine at any time, material ID on the station and its processing parameters are dynamically displayed to help the administrator to monitor the status of material handling and the processing status of a machine in the production process. The procedure of data visualization is presented in Figure 7-17, in which the XML file is applied to implement real-time data

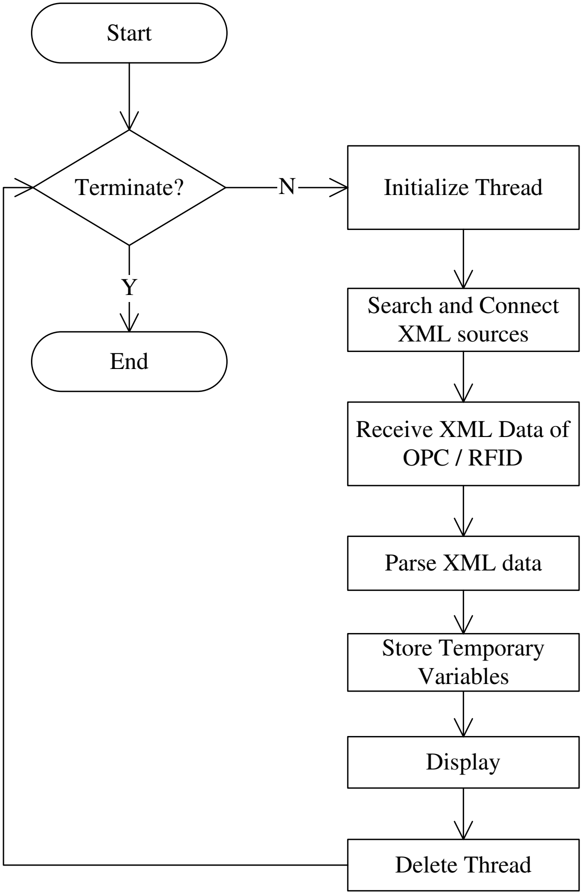

Figure 7-17 Workflow of real-time data visualization.

exchange in the system. In the visualization process, the production process Agent sequentially receives the data related to machines and material released by XML sources so as to resolve XML and to display the data.

7.4.2 Differential Manchester Encoding Rule-Based Warning Management

A differential Manchester encoding rule-based incident response mechanism is developed by the warning management Agent to judge the abnormal status of machines. The Manchester encoding technique proposed in 1949 is a synchronous clock encoding technique, which is used to encode a synchronous bit stream clock and data by the physical layer.[5] The Manchester encoding technique is applied in the LAN data transmission. In the Manchester encoding technique, there is a hopping in the middle of each bit, and the intermediate hopping can be regarded as either clock signal or data signal. Changing from low level to high level indicates a logic value '0'; while changing in the opposite direction indicates a logic value'1'. The differential Manchester encoding technique has been developed to improve its encoding technique. In the differential Manchester encoding technique, the hopping in the middle of each bit only provides clock timing; a logic value '1' indicates that the polarity of the signal bit is the same as the previous value; while a logic value '0' indicates that the polarity of the signal bit is different from the previous value, as shown in Figure 7-18.

Since there are fewer changes in differential Manchester encoding rule than in Manchester encoding rule, it is more suitable for high-speed transmission of information. In the production process of manufacturing systems, the production tracking and tracing system is required to respond rapidly to sudden changes in the actual manufacturing environment. In particular, when abnormal production situations occur, an alert should be issued immediately by the tracking system. The system is required to respond to changes in events, rather than data transmission; its focus is on the changing moments of events. Therefore, the

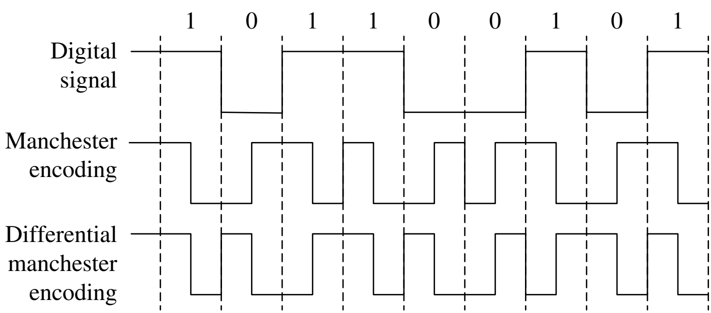

Figure 7-18 Manchester encoding rule.

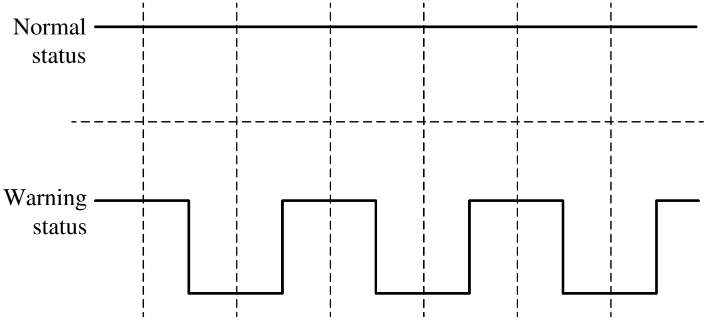

Figure 7-19 Status of differential Manchester encoding rule.

differential Manchester encoding rule is consistent with the abnormal response mechanism of the production process. In the production process of manufacturing systems, the production tracking and tracing system is adopted to monitor status changes of operators, machines and quality. A logic value '1' indicates that operating parameters are normal; while a logic value '0' indicates that operating parameters are abnormal. Its corresponding differential Manchester encoding rule is presented in Figure 7-19. When operating parameters are normal, it is encoded as a single high-level signal; when operating parameters are abnormal, it is encoded as alternating high and low signals, and the system issues an alert simultaneously.

By comparing the normal signal with the warning signal, it is noted that it almost operates normally in the production process, in which the signal is constant; while abnormal events seldom occur with changing signal output. In addition to machine abnormalities in the shop floor, there are material abnormalities and other abnormalities, as shown in Figure 7-20. The warning management Agent is used to determine whether the material is lost-based on the time difference to obtain the material information between two adjacent stations. After reading the material information, the former station sends the code to the latter station; if the latter station doesn't obtain the material information within a specified period, then it is believed that the material is lost; otherwise, the latter station will send the material information to its subsequent station, and the information is transmitted in such a way until the end of the production line. The material

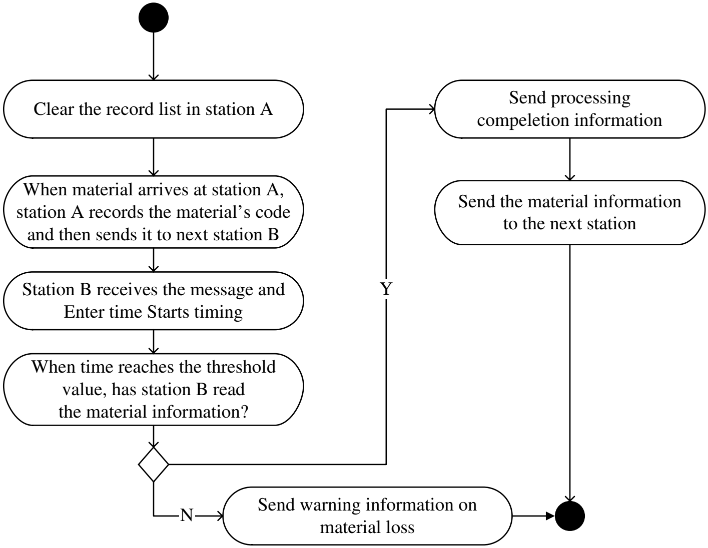

Figure 7-20 Material loss prevention algorithm.

loss monitoring system of the production line is not applied in product quality control stations; material with quality problem will be offline, but this doesn't mean that the material is lost. Hence, this algorithm doesn't contain the inspection station and the product repair station while setting the middleware. When the product is offline due to quality problems, the middleware will remove this product's information from the incoming list before going offline in order to prevent the false alarm.

7.4.3 Material Identification Technology for Production Process Tracking and Tracing

The process task is used to integrate the information obtained in the production process tracking and tracing Agent, such as material, machine, quality, operator and so on. Process production planning is the foundation of all the information. The information involved in production planning related to material, machine, quality and operator is obtained by the tracking system at the initial stage of the production process. The information obtained in the assembling production process related to material, machine

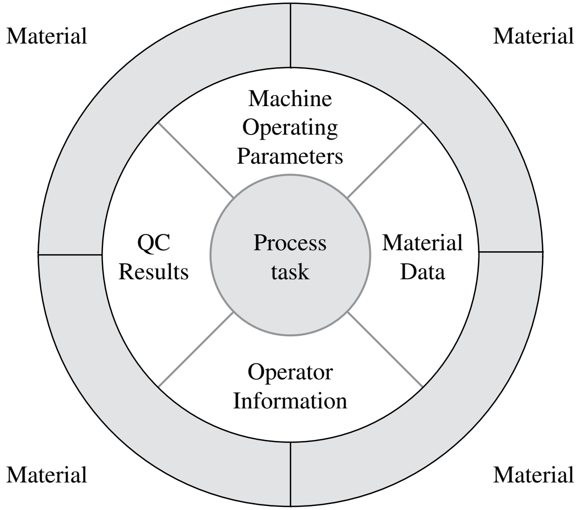

Figure 7-21 Tracking information.

production parameters, product quality that is acquired by quality inspection devices and operators is recorded through real-time data acquiring technology. All the information is integrated by the material information involved in the production process in order to track the production process information in real time. When the production processing process is completed and the product is offline, all the information related to the product is associated with the product data so as to achieve traceability of the production process, as shown in Figure 7-21.

The fundamental technology of production process tracking and tracing is material identification and tracking technology. Hence, the research on material-based information relevance is the foundation of the research on production process tracking and tracing information integration and production process visualization. The research on the production process tracking system will focus on implementing the relevance of information related to material, machine, product quality and operator. Since various kinds of information obtained in the production process are collected in real time, it is possible to develop a data relevance mechanism based on time. In this section, the association of material information and machine information based on time is used to illustrate information relevance algorithms involved in the production tracking system.

The Multi-Agent material data-acquiring system is adopted to read, store and issue material information involved in the production process, and to store machine operating parameters in the database. Apart from the fundamental information, there is need to consider time information involved in the data-storing procedure, the association of material information and machine information is achieved on the basis of this uniform time information. The material label is used to record the time that the material enters or leaves the station in the effective working period; however, this record usually doesn't match the operation processing time for processing the material recorded by the processing machine at this station. As a consequence, there is need to validate whether the material record matches the machine record; the verification method is presented as follows:

$$\lambda_1 = \frac{\frac{(T_{b1}+T_{b2})}{2} - \frac{(T_{a1}+T_{a2})}{2}}{T_{a2}-T_{a1}}, \lambda_2 = \frac{\frac{(T_{b1}+T_{b2})}{2} - \frac{(T_{a1}+T_{a2})}{2}}{T_{b2}-T_{b1}} \tag{7.1}$$

where

λ_1 denotes the approval parameter 1 at the station;
λ_2 denotes the approval parameter 2 at the station;
T_{a1} denotes the starting time of the processing machine record at the station;
T_{a2} denotes the completion time of the processing machine record at the station;
T_{b1} denotes the starting time of the material record at the station;
T_{b2} denotes the completion time of the material record at the station.

Since the machine's working area is greater than its corresponding station area at the station, the time range of each material record is generally greater than that of machine production parameters recorded by the processing machine, i.e. $\lambda_2 \le \lambda_1$. After the above processing procedure, equation 7.2 is used to determine whether the material record matches the machine record at the station.

$$\lambda_b \le \lambda_2 \le \lambda_1 \le \lambda_a \tag{7.2}$$

where

λ_b denotes the boundary material parameters at the station;
λ_a denotes the boundary machine parameters at the station;

Due to the setup difference between different machines at different stations in the same line or at the same stations in a different line and different operation processing times required in different machines and other factors, λ_a and λ_b are not unified. Hence, it is necessary to determine them by experiments after completing the machine setup process.

As in the previous presentation, the relevance verification algorithm based on material identification and its corresponding machine processing parameters in the production process is outlined in detail. Since product quality parameters are acquired and stored by the online quality inspection device, quality information corresponds to operating parameters information on the quality inspection device. Consequently, the association of material information and quality information is achieved as the material data is correlated to the production machine data. Similarly, the association of material information and operator information at its corresponding station can be achieved. Therefore, relevant technologies of material information and machine information, operator information and product quality information involved in the production process are proposed in order to provide a foundation for the production management system to track production status, treat dynamic events, and inquire and backtrack production parameters for products.

Another important function of the production process tracking and tracing Agent is to trace the products' information. The production process history information inquiry consists of multiple parts: machine-based production information inquiry, material-based production information inquiry, operator-based production information inquiry and quality-based production information inquiry. These inquiries are closely related to the production process tracking process; they are generated by reversing this production process tracking process. The inquiry algorithm is presented in Figure 7-22. In the procedure of the production process history information inquiry, firstly, the system obtains the record lists specifying the information related to material, machine, quality and operator, and then

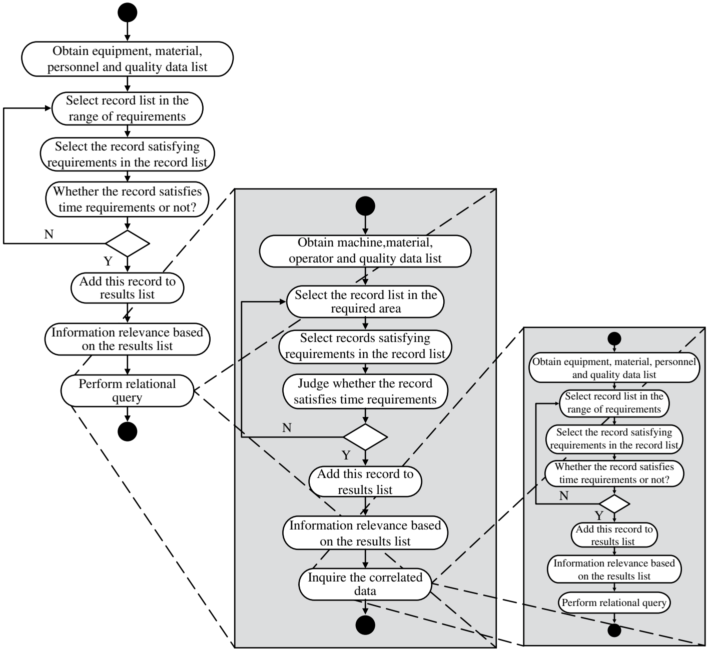

Figure 7-22 The history information inquiry algorithm in the production process.

checks these lists one by one, and eventually generates a required history information list.

Different from the production process history information inquiry, the content acquired by product backtracking is divided into batches, and its focus is on the quality, origin of material and traces of finished products in the batch. After each batch is completed, the system statistically arranges this batch data, and the preserved information related to operator, material, machine and quality should be unified in association and classification and saved in batches according to their requirements. In addition, the track information record is updated after products' output.

With reference to the history information inquiry algorithm in the production process shown in Figure 7-22, the product backtracking inquiry and recall operation algorithm is developed by the system on this basis. Since the product information is classified when the processing product is completed, and product backtracking process needs to be implemented with a lower requirement of the detail information, the complexity of 'correlated information' in its corresponding algorithm will decrease.

In Figure 7-22, the data layer consists of data related to material, machine and operator involved in the production process tracking and tracing process. The data processing processes are classified as those based on real-time data tracking and those based on historical data backtracking. Since XML is used to transfer the real-time data, the data processing process based on real-time production data tracking is implemented on the basis of data analysis in order to match various types of data and display these data in the interactive interface. The historical data is collected from the database, the quality data included in the historical data is validated by the quality standards before displaying it. In the display process, based on the backtracking information, inquiry methods are classed as those based on materials and those based on machine; the final result for any inquiry way is displayed on the interactive interface on the basis of correlated data.

When implementing the production process, the aim of the production control process is to track resources related to machines, materials operators involved in the production process, and to complete production tasks in accordance with the pre-requirements of conditions related to quantity, time, quality and cost and so on. It is necessary to accurately perceive and rapidly solve various abnormal events that have occurred in the production period. In this process, multiple Agents in the production control system interact with each other, and complete the production control process by collaboration.

In the production control process, the most frequent interaction among Agents happens in the production process tracking and tracing process. The production process tracking and tracing Agent is required to interact with other Agents, and to store the obtained information after processing in the production process tracking and tracing process, as shown in Figure 7-23.

Firstly, the production process tracking and tracing Agent is used to collect production process–related real-time data to the Multi-Agent material data acquiring system, Multi-Agent machine data acquiring system, and the quality management

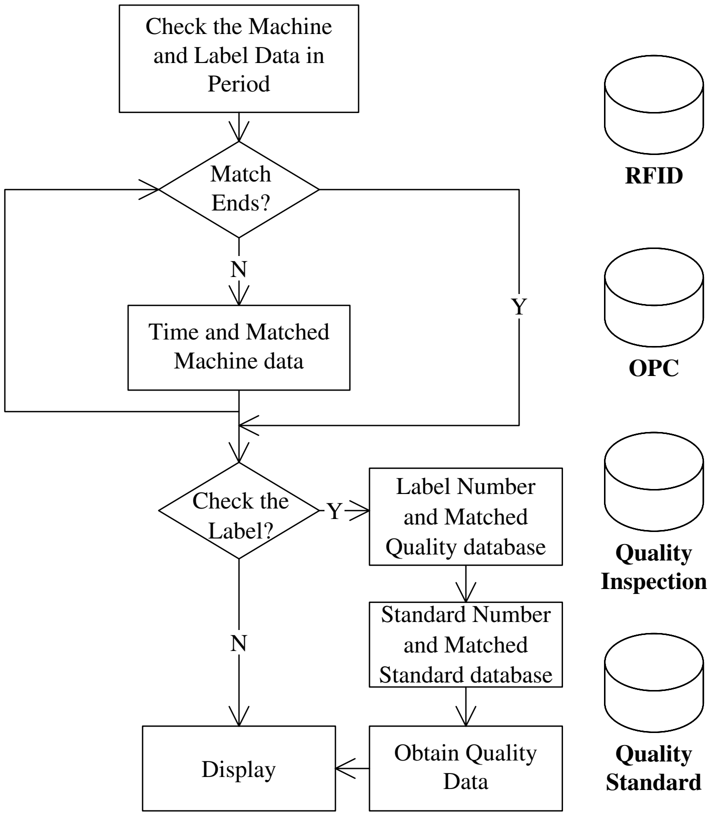

Figure 7-23 Inquire about production data and issue backtracking production orders.

Agent, and then to associate these data, and to store them in the database. When a warning occurs in the production process, the production process tracking and tracing Agent will also correlate the warning information with the material and quality information of its corresponding machine. Finally, the production process tracking and tracing Agent will analyze production performance data as a part of the tracking process.

In the procedure of production process monitoring, the status of task orders is updated, which will trigger further updates related to the production plan status and the product order status. The interaction among them is illustrated in Figure 7-24.

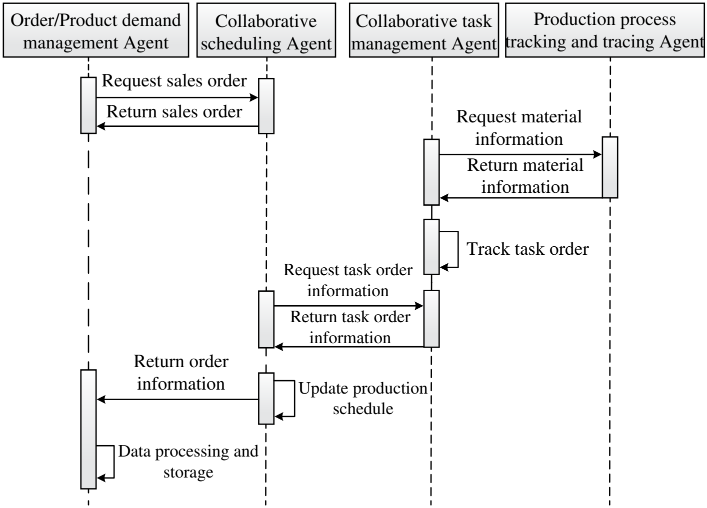

Figure 7-24 Interaction among Agents in the order management process.

First, the production process tracking and tracing Agent is used to acquire real-time parts processing information, and to update the task orders' status. Next, when the tasks on the task order are completed, the collaborative task management Agent is adopted to update the real-time status of the production plan. Finally, the final status of the production plan will affect results of sales orders in display and storage.

7.5 Conclusion

First, this chapter has introduced the architecture of a Multi-Agent production control system. Next, a detailed description of several important business Agents has been presented. These Agents included production monitoring Agents, warning management Agents, production performance analysis Agents, production process tracking and tracing Agents, collaborative task management Agents, quality management Agents and so on. Finally, technologies to achieve key businesses of a Multi-Agent production control system have been explained in detail.

References

[1] Guochao Yin. Study on Scheme and Application of RFID Based MES in Discrete Manufacturing Industry. Master's Thesis, Chongqing University, Chongqing, China, 2008, 22.

[2] EAN Belgium Luxembourg. EAN UCC Standard for Shared Supply Chain Traceability, 2003.

[3] APICS. APICS dictionary [EB/OL] [2012-02-22], http://www.apics.org/gsa-main-search.

[4] ISO22005:2005. Traceability in the Feed and Food Chain-General Principles and Guidance for System Design and Development, 2007.

[5] Forster, R. Manchester Encoding: Opposing Definitions Resolved. *Engineering Science and Education Journal*, 2000, 9(6): 278–280.

8

Multi-Agent-Based Material Data Acquisition

8.1 Introduction

The pull production control process in manufacturing systems is required to collect real-time data, and the collected material data is transferred to the production planning and scheduling layer by using some related technology.[1] During this procedure, the material data acquisition and the equipment data acquisition are the most important parts. This chapter explains how to collect the material data in manufacturing systems. Since the advantages of RFID technology include real-time reading and writing, high recognition efficiency, full marks and precise identification, a Multi-Agent RFID-based material data acquisition system has been developed.[2]

8.2 RFID Technology

8.2.1 Development of RFID Technologies

Radio Frequency Identification (RFID) is a non-contact automatic identification technology. Its basic principle is to transfer data between a reader and an RFID tag by using radio frequency signal transmission characteristics and spatial coupling (inductive or electromagnetic coupling) data transmission characteristics.

The implementation of RFID technology consists of three parts: an RFID tag, a reader and an antenna (as shown in Figure 8-1). For some RFID equipment with less power demand,

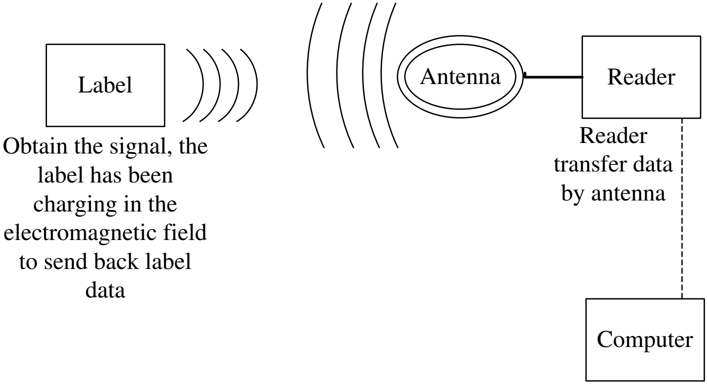

Figure 8-1 RFID

an RFID reader together with an antenna is referred to as a reader. When an object attached to an RFID tag moves into the reader-driven antenna range, the contactless reader can capture it and read out the data in the RFID tag, which enables wireless item identification within a certain range. Some RFID equipment with writing functions are able to write useful data into the attached tag when the object is within the readable range.

As early as World War II, RFID technology has been used by the U.S. military to identify enemy and allied planes; it was an automatic identification technology evolved from radar. In 1948, Harry Stockman published a paper "Communication by Means of Reflected Power" in the IEEE Radio Frequency Integrated Circuits Symposium, which provided the theoretical foundation of RFID technology[3]

In the 20th century, theory and application of radio technology are among the most important scientific and technological development achievements. Developments of RFID technology can be divided into the following stages.[4]

1941–1951: Improvement and application of radar-spawned RFID technology. In 1948, "Communication by Means of Reflected Power" provided the theoretical foundation of RFID technology.

1951–1960: RFID technology early exploration stage, mainly in the laboratory.

1961–1970: Theory of RFID technology has been developed and some applications have been launched.

1971–1980: RFID technology and product development entered a great development period. Various applications have appeared.

1981–990: RFID technology and products came into the commercial application stage.

1991–2000: Standardization of RFID technology is being increasingly considered. RFID products are widely used and have become a part of our lives.

2001 until now: More and more RFID products such as active RFID tags, passive RFID tags and semi-passive RFID tags have been well developed.

The cost of an RFID tag continues to decrease, while the scale of industrial applications expands. RFID technology theory has been enriched and promoted. Single-chip RFID tags, multiple RFID tag reading, wireless readable and writable, passive RFID tag remote identification, RFID tag fast-moving object identification have all been developed. In fact, the world's biggest retailer (Wal-Mart) has announced that RFID technology will be used in widespread applications, and the U.S. military has announced that RFID technology will be used to identify and track military material, which will greatly promote the research and application of RFID technology.[5]

With the development of RFID technology, its products are diverse. RFID equipment mainly includes:

1) Active RFID tags and passive RFID tags according to different power supplies

   Active RFID tags use the energy from batteries in the tags, have an identification range up to ten meters, or even hundreds of meters. However, their lives are short, and prices are high; they are generally too big to be easily attached. Passive RFID tags do not have batteries; they use coupled electromagnetic energy emitted by the reader as their power. The advantages include light weight, small volume, long life and low price. However, the effective range is restricted, normally from ten centimeters to ten meters, and large reader-transmitter power is required.

2) Active, passive and semi-passive RFID systems according to the data modulation type

In general, RFID tags without inner batteries are passive RFID systems, while RFID tags with inner batteries are active RFID systems. Active RFID systems send data to the reader by using their own RF energy. The data modulation types include amplitude modulation, frequency modulation or phase modulation. Passive systems transmit data by using modulation mode scattering, which have to use a carrier of the reader to modulate their signals.

In active systems, RFID tags use the energy from batteries in the tags; therefore they are highly reliable and have a long identification range. In passive systems, RFID tags without any inner power supply rely on outside energy. Typical apparatus for generating electrical energy are coils and antennas. When the RFID tag enters the working area of the system, the electromagnetic waves received by the antenna coil will generate a specific induction current. When the RFID tag goes through the rectification circuits, the micro switch will be activated, and the RFID tags will be supplied with the electrical power.

Semi-passive systems are also called battery-supported repercussions-scattering modulation systems. Semi-passive systems have their own power batteries which provide electric energy for internal digital circuit in RFID tags. RFID tags do not take the initiative to send data by using their own energy until they are activated by the reader's energy field. The data modulation type is backscatter.

3) LF, HF and UHF and microwave systems according to operation frequency

   The frequencies that are used to send signals by a reader are the operation frequencies of an RFID system. They can be divided into five main areas: LF (30–300 kHz), HF (3–30 MHz), UHF (300 MHz–3 GHz) and microwave (higher than 2.45 GHz). LF systems work at the frequency of 100–150 KHz, among which 125 kHz and 134.2 kHz are normally used. HF systems work at the frequency of 10–15 MHz, among which 13.56 MHz is normally used. UHF systems work at the frequency of 850–960 MHz, and 915 MHz is the common one. Microwave systems work at the frequency of 2.4–5GHz.

4) Read-Only, Read-and-Write, Write-Once-Read-Multiple times

   The RFID tags can be divided into three different types according to different internal memories: the read-and-write

(RW) card, the read-once-read-multiple times (WORM) card, and the read-only card.

5) Identification tag and portable data file according to data storage capacity

   For an identification tag, a number or multiple numbers, letters, or strings are stored in the RFID tag for the purpose of identification or entering the database of an information system. An identification tag only stores the sign number to identify people, things and places. More information about the identified project can be searched in its corresponding database.

The portable data file indicates that the data stored in the RFID tag is too much to be treated as a data file. These kinds of RFID tags are normally programmable. In addition to the sign number, other related information about packing instruction and process description is stored in RFID tags.

8.2.2 RFID Technology Standard

Currently there is no consensus standard on UHF RFID technology. International standards[6] include the European and American Electronic Product Code (EPC), the Japanese Ubiquitous ID (UID), the ISO 18000 series, and the South American iP-X.

1) *EPC standard*. EPC global is responsible for maintaining the EPC standard. In 1999, the Massachusetts Institute of Technology (MIT) established the Automatic Identification (AUTO-ID) Center, and proposed the EPC concept. Then, EAN.UCC officially took over the global dissemination and application of the EPC standard and established EPC Global. The Auto-ID Center has been renamed as the Auto-ID Lab, and the research work of EPC is promoted by the Auto-ID Lab.[7] EPC global is a joint venture of the European Article Numbering Association (EAN) and the Uniform Code Council (UCC). It is a non-profit organization entrusted by industry companies and is responsible for global standardization of the EPC network, in order to identify the product in supply chains more quickly, automatically and accurately.

Except for the definition of an RFID tag, EPC global has defined the structure of a reader in its fifth document. In the sixth document, EPC global defined the structure of a savant (later renamed Application Level Events (ALE)), which is responsible for collecting and storing EPC information sent by an RFID reader and taking corresponding actions. The seventh document defined the PML (Product Markup Language), which is an XML-based format language as a standard for communication between EPC hosts. The structure of an ONS (Object Name Service) is defined in the eighth document. Through the network searching mechanism of an ONS, users can share information with each EPC group: for example, the name of the product, specifications, manufacturing date, and so on. Its function is similar to a DNS in the network environment. All relevant documents are currently in ongoing updates.[8]

2) *UID standard*.[9] The UID standard is published by the Ubiquitous ID Center. Ubiquitous ID was founded in March 2003, as a combination of many Japanese-related associations and organizations. The UID standard proposed the encoding scheme of a UID code, in which the length of a UID code is 128 and can be extended to 256, 384 or 512 as needed. Compared with EPC standard, UID standard is more complete because UID standard contains definitions of transition practices, tags, and other parts.
3) *ISO 18000 series of standards*. In terms of contactless smart cards, the current technical specifications developed by ISO focus on the communications protocols. They include the ISO l8000 series for supply chains, ISO 10536 for Close Coupling Cards, ISO14443 for Proximity Cards, and ISO l5693 for Vicinity Cards, among which ISO 14443 and ISO 15693 are commercially popular.

The EPC specification is led by American EPC Global Association, which includes Wal-Mart, Tesco and other companies, and it is supported by IBM, Microsoft, Philips, Auto-ID Lab. UID standards are mainly maintained by Japanese manufacturers. ISO organizations currently developed ISO18000-6 protocols for RF at UHF, and EPC global developed standard Electronic Product Code for UHF. These two organizations for standardization have been converging recently. ISO examines and approves EPC Gen 2 Class 1 UHF, modifies it in order to generate a 18000-6 RFID interface standard for 860–960 MHz ISM equipment.

In terms of RFID coding, the EPC global standards are compatible with the bar coding standard widely used in mixed model manufacturing systems. In terms of air interface, the frequency of RFID equipment applied in mixed model manufacturing systems is mainly in the UHF frequency band.[10] As a member of EAN.UCC, the Chinese Article Numbering Center (ANCC) is also actively involved in promoting the EPC standard. The ultimate goal of the EPC standard is to establish a global open standard for every product. It consists of six areas:[11]

8.2.2.1 EPC Coding Standard

The EPC code is a new coding standard, which is compatible with EAN/UPC code. In an EPC system, EPC code is combined with GTIN instead of replacing the existing barcode standard. In the future, current barcode standards will gradually transfer to the EPC standard, and the EPC standard may coexist with EAN.UCC system in the future supply chain. The EPC code segment is assigned by EAN.UCC. In China, GTIN code in EAN. UCC system is allocated and managed by the Chinese Article Numbering Center. Moreover, ANCC is about to start EPC services to meet the demands of domestic enterprises.

EPC code consists of a version number and three sections of data (i.e., domain manager, object category, serial number). The version number is used to identify the EPC version, which indicates the length of the subsequent code segment. The domain manager describes the information of the related manufacturer, for example, Coca-Cola Company. The object category records the product information about its exact type, for example, 330 ml U.S Diet Coke (a new product of cola)". The serial number is used to uniquely identify the product. The EPC encoding scheme is shown in Table 8-1.[12]

8.2.2.2 EPC Tag

An EPC tag consists of four parts: antennas, integrated circuits, joint of antennas and integrated circuits, and the bottom for

Table 8-1 EPC encoding scheme.

		Version	Domain Management	Object Category	Serial Number
EPC-64	TYPE I	2	21	17	24
	TYPE II	2	15	13	32
	TYPE III	2	26	13	23
EPC-96	TYPE I	8	28	24	36
EPC-256	TYPE I	8	32	56	160
	TYPE II	8	64	56	128
	TYPE III	8	128	56	64

hosting the antennas. A 96 or 64-bit EPC code is the only information stored in the RFID tag. The high cost of an EPC tag is one of the biggest obstacles to promoting the technology on a large scale. Therefore, the cost of an EPC tag is being reduced in order to allow the EPC tag to play a role in single product tracking. The following measures have been taken into consideration: to reduce the chip, to develop new antennas and to look for alternatives to silicon.

In view of the functionality level, EPC tags can be classified into five categories. Currently the Class1 Gen2 tag is used for EPC experiments.

8.2.2.3 Reader

Readers use a variety of ways to exchange information with tags. Inductive coupling is the most common method to close read passive RFID tags. A magnetic field is formed between the antenna of coiled readers and the antenna of coiled RFID tags when they are close. RFID tags use this field to send electromagnetic waves to readers. The returned electromagnetic wave is converted into data information, which is the EPC code of an RFID tag. A reader costs about $1,000 or more, and most can only read information in a single frequency chip. In general, the HF RFID tag has a longer reading range. A typical LF RFID tag is readable only within a foot; while a UHF RFID can be read at a distance of 3.05–6.10 m.[13]

The operating frequencies of RFID systems can be divided into four main areas: LF (30–300 kHz), HF (3–30 MHz), UHF (300 MHz–3 GHz), and microwave (higher than 2.45 GHz). LF RFID systems are normally applied in short-distance, low-cost applications, for example, most access-controls, charge cards and so on. HF RFID systems are common applied in library management, medical logistics systems and so on. UHF RFID systems are implemented in logistics, production automation, railway parcels and other fields. Microwave is used for scenarios requiring longer reading ranges and higher reading and writing speeds. Its antenna beam direction is narrow, and the cost is relatively high; therefore it is normally applied in railway monitoring and highway toll systems.[14]

8.2.2.4 Savant (a Neural Network Software)

After adding RFID tags onto each product, the reader will receive a series of EPC codes in the processes from production, transportation to sale. In this procedure, the most important and the most challenging part is how to transfer and manage the data. The Auto ID Center has developed software named Savant, which is a neural system of the new network.

Savant uses a distributed architecture to hierarchically organize and manage the data flow. Savant will be applied in stores, distribution centers, regional offices, factories and even on truck or cargo aircraft. A Savant system at each level will collect, store and process the information, and communicate with other Savant systems.

8.2.2.5 Object Naming Service (ONS)

The Auto-ID Center considers that it is necessary to adopt some special network structures for an open, global tracking goods network. Rather than storing EPC codes in the RFID tags, some matching methods are still needed to match the EPC code with the corresponding product information. This function is realized by Object Naming Service (ONS). When a reader gets the information stored in an RFID tag, an EPC code is transferred to a Savant system. The Savant system finds the location where the product information is stored by using ONS on a LAN or the Internet. ONS indicates the server that stores the product data for the Savant system; the file will be found, and all information in this file will be extracted for managing supply chains.

8.2.2.6 Physical Markup Language (PML)

An EPC code is used to identify a single type of product; all useful information related with the product is described by a new standard computer language – Physical Markup Language (PML). PML is developed from the well-known Extensible Markup Language (XML). Since it has become a unified standard to express all natural objects, processes and the environment, PML will be applied widely and enter all industries. The purpose of the AUTO-ID center is to start from a simple language and to encourage adopting new technologies. PML is still evolving to become a more complicate language, just like the basic language of Internet – HTML.

As in the previous presentation, an EPC standard system is a global Internet-oriented RFID information system. Its aim is to establish a unified RFID data network structure, and users can obtain the history information of the product in a savant system by scanning its RFID tag. The EPC codes, RFID tags and readers are the most essential parts in product tracking systems. In terms of EPC codes, EPC-64 and EPC-96 are the most common standards. In terms of RFID tags, the most popular ones are RFID tags obeying Class 1 Gen 2 standard. RFID readers are mainly compliant with the air interface standard at 13.56 MHz and UHF.

8.3 Agent-Based Material Data Acquisition System

8.3.1 Requirement Analysis of Material Data Acquisition

The requirements for material data acquisition process in manufacturing systems consist of the following three aspects:

1) *Real-time capturing material data*. The production planning and control process is required to collect, process, upload and download the material data in real time.
2) *Integrating heterogeneous device information*. The material data acquisition system should provide data in a uniform format in order to read and feedback information. For various types of data acquisition equipment deployed on the shop

floor, a method to shield the differences in hardware should be adopted to obtain the uniform data.
3) *Integration*. The real-time communication between a Multi-Agent production planning and control system and a Multi-Agent material data acquisition system is required to ensure the effectiveness of making decisions in the dynamic environment.

8.3.2 Multi-Agent RFID-Based Material Data Acquisition Structure

Functions and physical mapping methods of Agents were presented in Chapter 3. In terms of RFID technology, physical decomposition is adopted in order to generate independent defined Agent objects, which include the RFID reader Agent, the RFID tag Agent and the RFID middleware Agent. As shown in Figure 8-2, these Agents constitute the Multi-Agent material data acquisition system, so as to provide the following functions:

1) *Hardware management*. A Multi-Agent-based material data acquisition system is used to manage RFID tags, readers and antennas and other equipment, in order to register the hardware in the software system, and to develop the RFID operating environment. Material data acquisition functions must be implemented by the interaction between multiple Agents. Since the data acquisition hardware interaction involves RFID, it is necessary to carry out the mapping process between the physical hardware and the software.

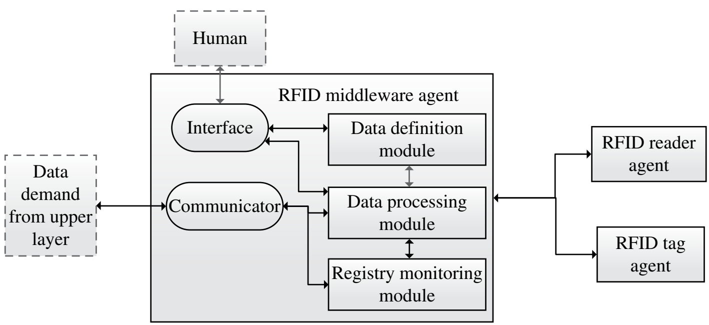

Figure 8-2 Multi-Agent-based material data acquisition system.

2) *Data acquisition and processing.* A Multi-Agent-based material data acquisition system is used to obtain real-time information about materials so as to manage heterogeneous material data acquisition equipment.

The data processing process transfers the data stored in an RFID tag Agent to the data in the B2MML (Business to Manufacturing Markup Language) format. The B2MML language plays a role in providing a data exchange standard for different levels of enterprise application systems in order to enable easier data exchange. The XML Schema is used to express standard defined resources and information flow, and define the exchange data in control systems. The data collected can be reused in order to share information between systems by using this specification conversion. The data processing module has two-way functions and can also transfer the upper layer data provided by the request command of the communication, and then sent to the RFID middleware Agent or product Agent. The data processing procedure is shown in Figure 8-3.

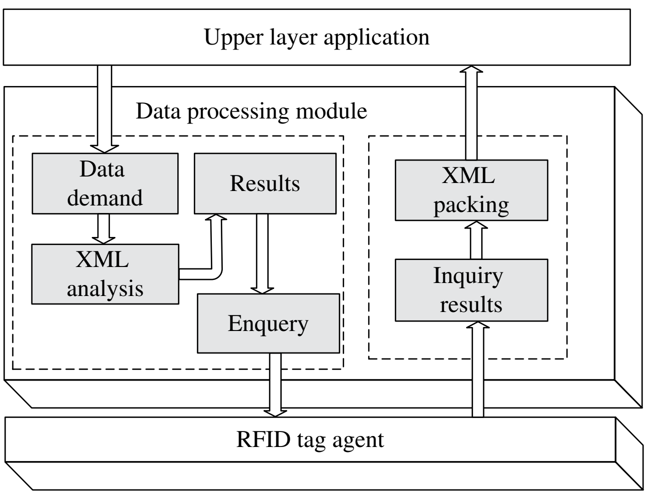

Figure 8-3 Data processing procedure.

The definition of data transmission is implemented by issuing XML. XML is applied to publish the core part of data stored in an RFID tag, which is defined as follows:

-

-

"tag coding record"
"equipment number"
"time for an RFID tag to enter the device reading area"
"time for an RFID tag away from the device reading area"
"the number of reading times for an RFID tag in the device reading area"

The data content contains the RFID reader ID number corresponding to an RFID tag Agent, the RFID tag data content, the entering and leaving time points of an RFID tag and the number of reading times for an RFID tag. An RFID middleware Agent releases RFID tag data by using a frequency, and changes of the data are monitored by the application layer to obtain the latest material data. In addition, the data requirements are also transmitted to the RFID tag Agent by using an RFID middleware Agent.

3) *Data upload and release.* The Multi-Agent RFID-based material data Acquisition system is integrated with a Multi-Agent production planning system and a Multi-Agent production scheduling system in order to improve the efficiency of information transmission. When the Multi-Agent material data acquisition system receives the request sent by the upper application, it is processed and sent to the corresponding Agent. The feedback is sent to the Multi-Agent material data acquisition system, and the Multi-Agent material data acquisition system will transfer the data and send it to the upper application.

8.3.3 The Running Model of a Multi-Agent Material Data Acquisition System

The running procedure consists of the following phases: RFID reader and tag hardware register and Agent deployment phase, the material data acquisition request and the response phase. There are some changes in the Multi-Agent environment, such as RFID middleware Agent service changes, lost connections between RFID readers and RFID tags, and so on – all of these need to be dealt with through the interaction between Agents.

A RFID reader Agent and an RFID tag Agent are used to install RFID readers and RFID tags, and connect Agents with hardware devices. Then the RFID reader Agent and the RFID tag Agent respectively transmit the message to the RFID middleware Agent. The RFID middleware Agent returns the confirmation message after receiving the message, which indicates that the RFID reader Agent and RFID tag Agent registration process is completed. The running model is shown in Figure 8-4.

The information processing procedure is shown in Figure 8-5. Firstly, the application interface receives instructions, and the RFID middleware Agent informs the RFID reader Agent to configure the hardware. Secondly, the reader interface sends a reading command to the hardware, and the hardware returns the data. Thirdly, the data is stored in an RFID tag Agent, and is sent to the RFID reader Agent and the RFID middleware Agent. Finally, the data is processed, and the results are sent to the application layer through the application interface.

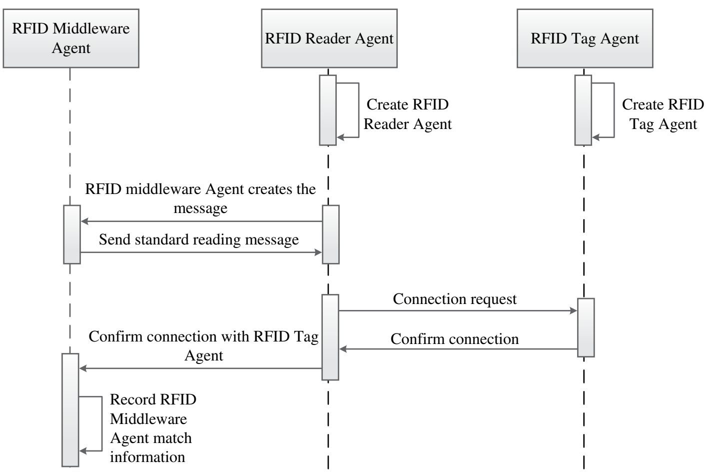

Figure 8-4 Agent installation and registration.

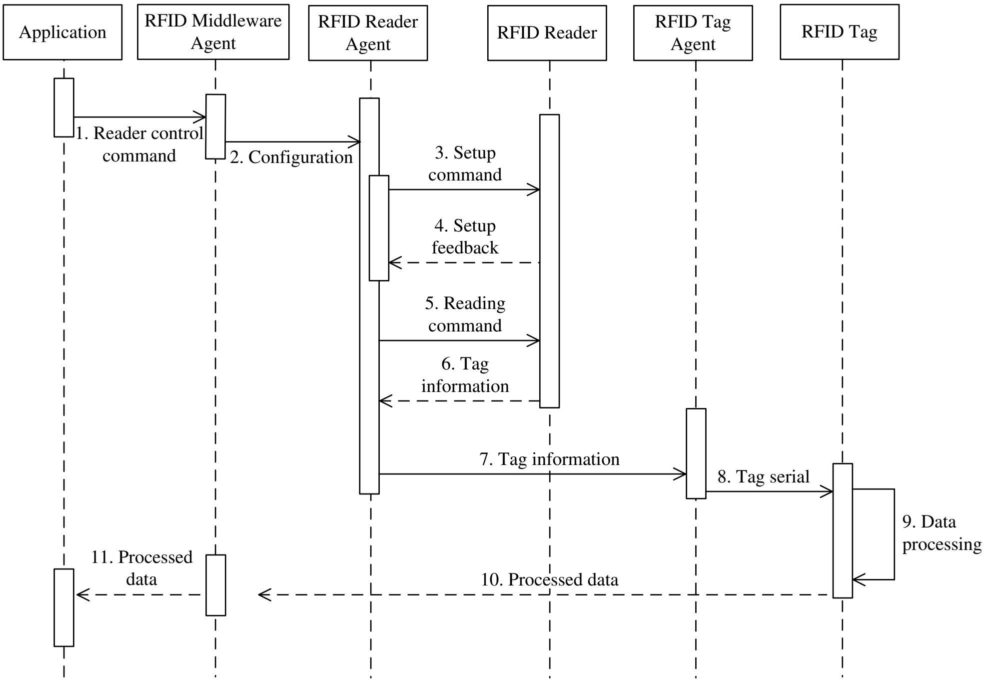

Figure 8-5 Information processing flowchart.

8.4 Agents in Multi-Agent RFID-Based Material Data Acquisition Systems

Multi-Agent systems play the core role for collecting material data; the upper production process tracking Agent obtains real-time material information in manufacturing systems by communicating with Multi-Agent systems. A Multi-Agent RFID-based material data acquisition system collects the material data by collaborating with an RFID reader Agent, an RFID middleware and an RFID tag Agent. The internal operation principles and external information interaction of Agents are presented in this section by using Agent structures and behavior models.

8.4.1 RFID Middleware Agent

An RFID middleware Agent is the core component of a Multi-Agent system as a coordinator Agent. The RFID Middleware Agent exchanges information with an RFID reader Agent in order to access and process material related data. At the same time, the RFID middleware Agent communicates with a production process tracking Agent to upload and release data in a production planning and control system. The behavior model of an RFID middleware Agent is shown in Figure 8-6.

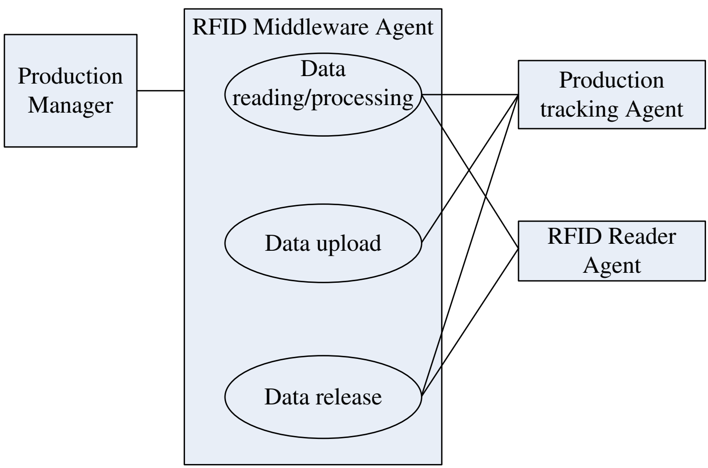

Figure 8-6 The use case diagram of an RFID Middleware Agent.

The RFID middleware architecture proposed by Auto-ID Lab contains a reader interface, a processing module unit, and the application program interface. Savant Specification 1.0 defines object name service (ONS) and other functions. Object Name Service is primarily used to determine production location when the product is traceable, and then to query the production status of the product. Due to complex manufacturing systems, diverse production processes, the MAS must be able to collect, store and publish data rapidly and accurately. Existing RFID systems require large amounts of data frequently exchanged between systems in order to enhance its versatility to meet the demands of various industries, which leads to a huge and increasing network overload. On the basis of Savant Specification 1.0 standard middleware architecture, this book combines production information records, inventory materials, and other demands in order to develop an RFID middleware Agent for manufacturing systems. The RFID middleware Agent is a unified Agent model, which consists of a basic definition module, an automatic driver module, an information processing module and an Agent communication interface. Its structure is shown in Figure 8-7.

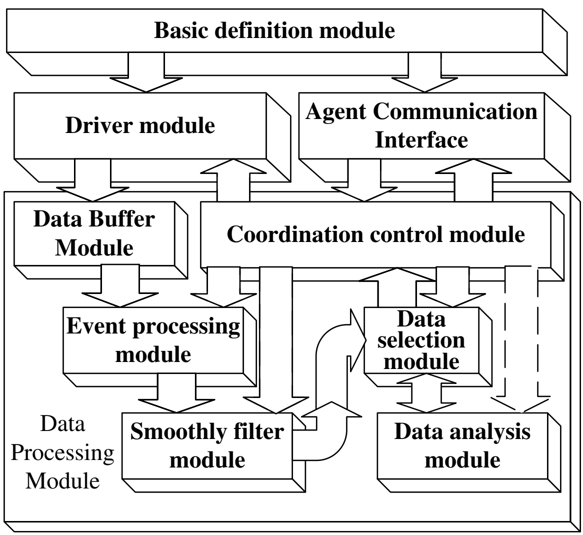

Figure 8-7 The Structure of an RFID Middleware Agent.

The first part is the basic definition module, which is used to define the underlying data. For the reader, it is necessary to define RFID frequency, communication pathways, reader location and other information; for RFID-tagged products, it is necessary to define their production information as well as data formats.

The second module is the automatic driver module, which is adopted to ensure that RFID equipment works normally in accordance with the basic definition module. The driver module completes data interaction between an RFID middleware Agent and a reader Agent, and its function consists of two parts: First, the module receives the command, controls the RFID reader and monitors readers. The second part is to receive product codes provided by the reader and send them to the information processing module. The automatic driver module is used to connect and test equipment, set operation parameters, collect and upload data. The reader interface module receives information defined by the basic definition module, converts the data to machine languages in order to set the operation parameters of the hardware device and send the reading command.

Since RFID technology standards are not uniform, EPC global developed Savant specifications to determine how to receive data for an RFID middleware Agent, and encourage hardware manufacturers to form a unified hardware interface. However, current hardware vendors often provide hardware API or simple interface programs, in which API or interface programs are rarely used to transmit data and send commands by using Savant definition. For RFID devices with common communication interface and API, the driver module offers standard USB, serial, and Wi-Fi interfaces. The Universal Device Interface is designed by using the EPC global standard (as shown in Figure 8-8) in order to achieve better compatibility between middleware and hardware. For the device with other third-party agreements or API, the driver module defines the device driver library to be called by these special devices. The middleware interface is primarily designed to complete six functions: connect readers, disconnect readers, set up readers, send reading instructions, stop reading and return RFID tags.

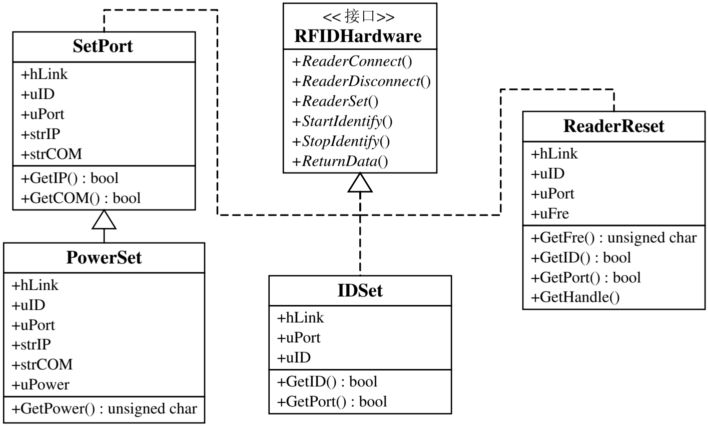

Figure 8-8 The reader interface of an RFID middleware Agent.

The third module is the Agent communication interface, which is used to exchange data between an RFID middleware Agent and other Agents.

The fourth module is the information processing center, which is the core of an RFID middleware Agent. Information received by readers goes through the buffer module and enters the information processing center. The information processing center sends the information to the related processing module in accordance with the requirements defined by the basic definition module. And the results are sent to the upper application by using the adapter. The information Processing Center consists of six processing modules: data buffer module, event processing module, smooth filter module, data selection module, data analysis module and coordination control module. The event processing module confirms the type of a new RFID tag, and then determines treatment methods; the smooth filter module transfers data to extract useful information by statistical classifications for other modules to call; the data selection module filters data in accordance with demands and sends the sorted data to other modules; the data analysis module is a backup module, its role is to optimize the data selection processing module in order to ensure information processing efficiency in complex situations. In addition to the above four core processing modules, there are two auxiliary modules, which have two functions: the data buffer and coordinated control.

The information processing center is the core of an RFID middleware Agent; four algorithms are investigated according to the status of the production manufacturing systems. The event processing module determines the type of an RFID tag. The smooth filter module organizes the data obtained by a reader to obtain valid data. The data selection module screens data according to demands. The data analysis module assists the data selection module to filter the data. Four algorithms are developed for four modules.

8.4.1.1 Event Processing

The event processing module is the gateway module of an information processing center. When a new RFID tag goes through the data buffer module and enters the information processing center, the event processing module needs to determine the properties of the RFID tag, and then to determine its handling method. Different types of events are dealt with by using different approaches. Event types are classified into the following three categories:

1) A new record to record the event that has not appeared in the middleware history;
2) A new record to record the event that is processed in the middleware;
3) A new record to record the event that appears after stops in the middleware history.

For a new record, the type of an event must be determined, then a treatment according to the event type is recorded, as shown in Figure 8-9.

When the event processing module obtains a new record, if there is no historical record of this RFID tag, it adds a new record; if the RFID tag has been processed in the middleware, it must further determine whether it has been out of the current event cycle. If the current event has left its cycle, the treatment is equivalent to a new record; while the old record of the same RFID tag does not depart from the current period, it is considered equivalent to the processing record in the current period, its post-processing mode is the same as the one of the processing record.

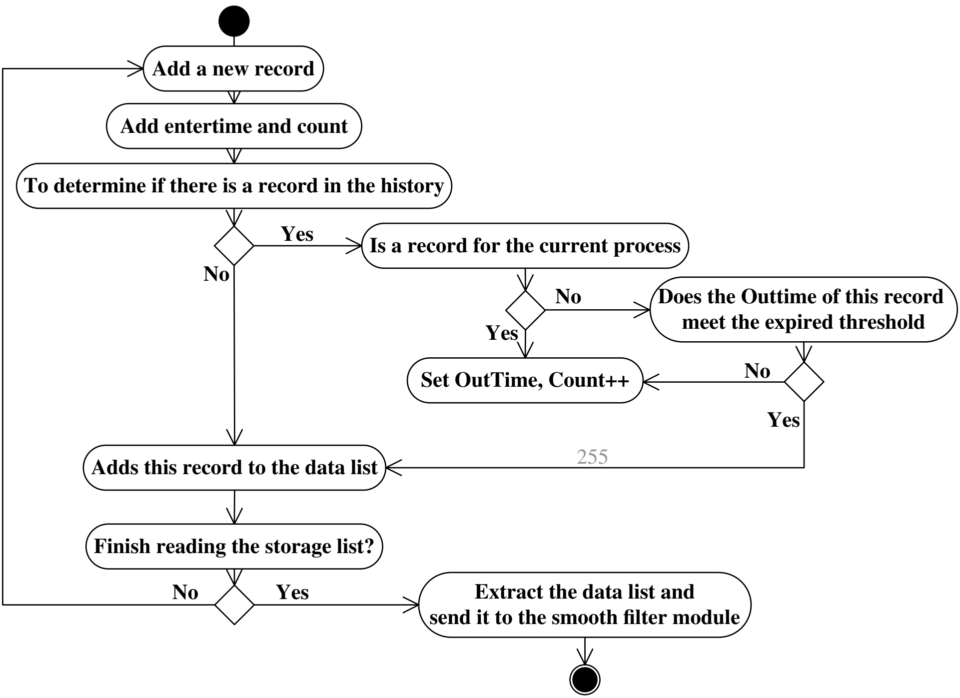

Figure 8-9 Algorithm of an event processing module.

When all records are processed, the event processing module generates a data list and sends it to the smooth filter module to determine the validity of each record.

8.4.1.2 Smooth Filter Data

The smooth filter module is the basic module of an RFID middleware Agent. The adjacent station distance of the manufacturing systems is usually less than 3 m, the material continuously transmits on the line, the material RFID tags of adjacent stations may conflict, and these RFID tags must be effectively distinguished. As the material will stay at a single station for a short period, the number of occurrences and the RFID tag can be recorded. The validity of the RFID tag information is judged by comparing it with the threshold value set in advance. The threshold value is related to the situation; it is set by the coordination control module according to the actual production situation.

As shown in Figure 8-10, the data storage list is provided by the information buffer module. The data stored in the information buffer module are read directly by each reader and are not processed. It includes this station's RFID tag data and other interference data. The middleware filters the data in both the time dimension and the space dimension in order to generate a valid data list. In the time dimension, the middleware obtains a particular residence time of an RFID tag at a station; the record outside of this period can be considered as independent of the station. In the spatial dimension, the reading effect of an RFID tag is related to distance. The closer an antenna, the better the reading effect; RFID tags beyond the readable range may be occasionally read; the number of times are necessarily much smaller than the closer ones, Therefore, the count threshold can be determined experimentally by the middleware; it indicates that the reading record below the threshold value is invalid. Information generates a record list through the smooth filter module, returns it to the information buffer module, and it will be stored in the database for other modules to call.

8.4.1.3 Data Selection

In most cases, only some of material or product is recorded rather than all the inventory. The information system obtains part of an RFID tag's information, for example, by period, according

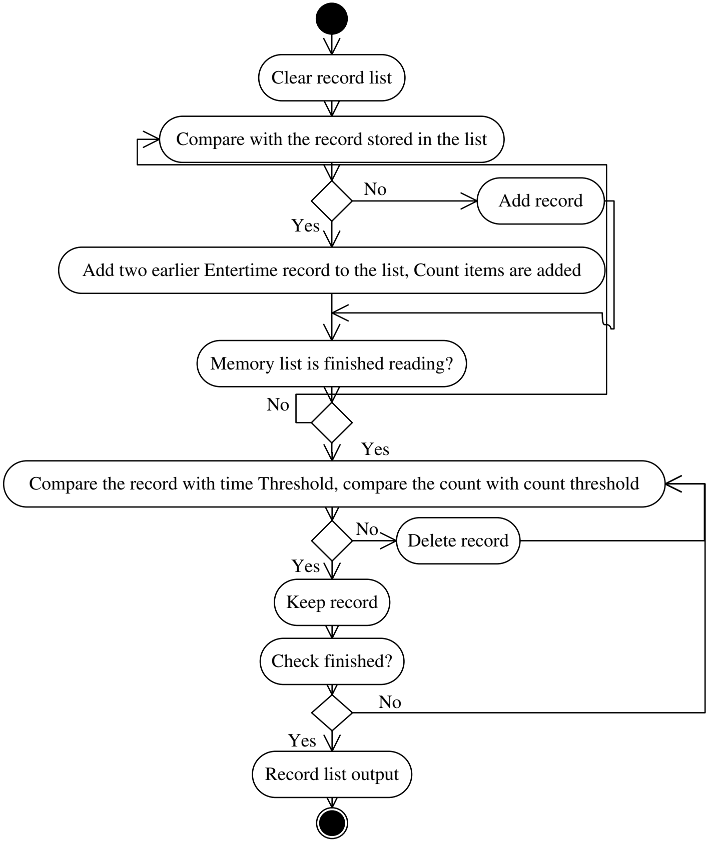

Figure 8-10 Flowchart of the smooth filter module.

to the station, according to the reader, by batch, press materials, and so on. All RFID-tag-associated data consist of two parts: reading time and RFID tag encodings. The data are selected in two directions according to the requirements of the information systems. For RFID-tag-encoded data, the selection process is made by setting the coding mask, and filtering the RFID tag scope of a mask in order to divide the RFID tag library into two parts: the inner mask and the outer mask. The data selection module is set up based on this demand.

RFID tag code bits are allocated according to the actual situation and EPC code bit allocation in order to generate the mask in the data selection period. For example, the code of a production line in the RFID tag encoding is needed in order to select all the RFID tag information related to the production line. The corresponding RFID tag information is obtained when the code queries the RFID tag library as a mask.

As shown in Figure 8-11, the data selection module obtains the input list, and the data list is compared with the mask set. If

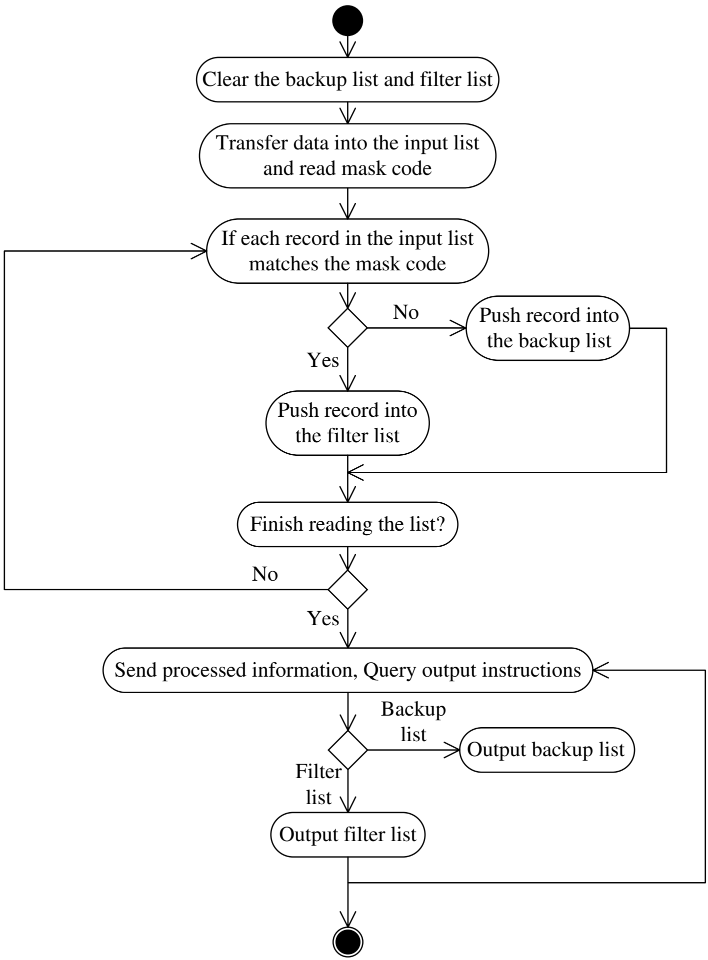

Figure 8-11 Flowchart of the data selection module.

the record satisfies the required range, the data is written into the output list. If it does not satisfy the required range, the data is written into the backup list. When all the data in the record list are inspected, the data selection module returns the output list and the backup list.

The data record list goes through the data selection module in order to generate a backup list and an output list. The data selection module is called by other systems and modules, and the module uploads the filter list or the backup list according to the system requirements.

8.4.1.4 Data Analysis

Data analysis is an auxiliary module of the data selection module. If the amount of an RFID tag data is very large and other systems in the application layer process the RFID tag data many times, then the large amount of RFID tag data will become a burden on the system. Large data transmission and computing will reduce the overall system efficiency or even make the system unable to meet production requirements. To deal with this situation, a data analysis module is developed to preprocess RFID tag data, which makes the latter data selection and data manipulation more convenient.

As shown in Figure 8-12, the length of EPC-96I Data encoding is 36 bits, if each encoding consists of three portions, then each RFID tag includes three data lists that store different data. Not all the bits in the material encoding appear for production tasks, only one segment changes many times, and the other two segments seldom change. However, the data processing process does not distinguish changes; all the RFID tag data are preprocessed. The data processing deletes duplicated secondary data in order to improve the efficiency of the data processing process.

Determine whether the A1 segment data is in a consistent relationship with the *i*th class in processing batch task (i = 0,1,2, …, n1).

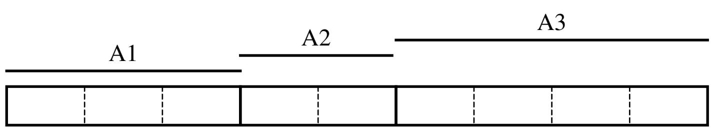

Figure 8-12 Tag encoding and data analysis.

A2, A3 segments follow the same rule. After the above operation, the RFID tag record is processed by an RFID middleware Agent as shown in Figure 8-13. The number of bits is shortened from 9 to $\lceil \log_{16} n_1 \rceil + \lceil \log_{16} n_2 \rceil + \lceil \log_{16} n_3 \rceil$. In the actual production process, the last segment often changes, all the records are changed to a "00A3" format in order to reduce the complexity of the value comparison. When all the data have been processed, the records are returned to the original coded data list by the inverse operation if it is necessary to show the significance of the encoded data.

The flowchart of an RFID middleware Agent is shown in Figure 8-13. The application interface receives instruction information from a reader, transfers it to the reader debugging command, and sends it to the appropriate reader by a reader Interface. At the same time, the application program interface sets the information processing process according to the requirements, and sends it to the information processing center. An RFID reader reads the RFID tag information, and sends it to the data buffer module through the reader interface. The information processing center calls the RFID tag information in the buffer module and sends it to the application program through an application program interface in accordance with the requirements.

8.4.2 RFID Reader Agent

An RFID reader Agent is used to manage RFID readers in manufacturing systems, and register hardware; while the RFID reader Agent is adopted to read and write data for an RFID tag Agent. Its behavior model is shown in Figure 8-14. The functions of an RFID reader Agent are simple, and are managed without complex business logic or knowledge; it is a reactive Agent.

8.4.3 RFID Tag Agent

The function of an RFID tag Agent is relatively simple (Figure 8-15); it is mainly responsible for receiving the data request sent by an RFID reader Agent and writing it in an RFID tag; while an RFID tag Agent is used to send material information stored in an RFID tag to an RFID reader Agent.

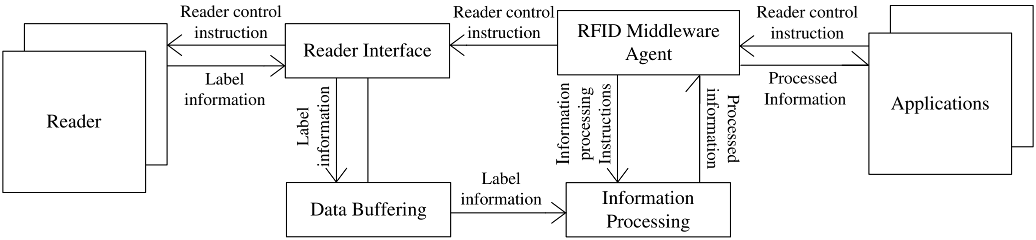

Figure 8-13 Flowchart of an RFID middleware Agent.

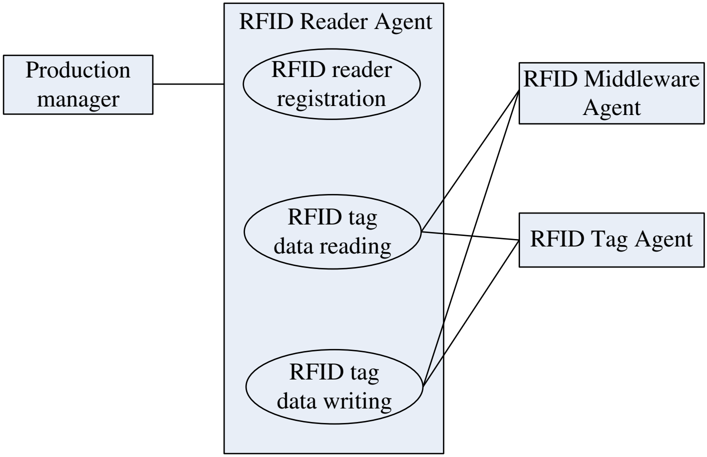

Figure 8-14 The use case diagram of an RFID Reader Agent.

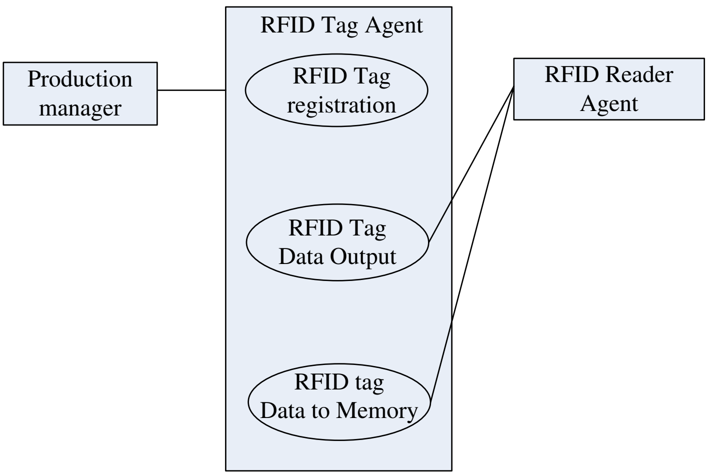

Figure 8-15 The use case diagram of an RFID Tag Agent.

The information transmitted by an RFID tag Agent is primarily material data information. The RFID tag Agent belongs to a reactive Agent.

Currently, since the storage capacity of an RFID tag has been improved, rather than the RFID tag ID, a lot of material

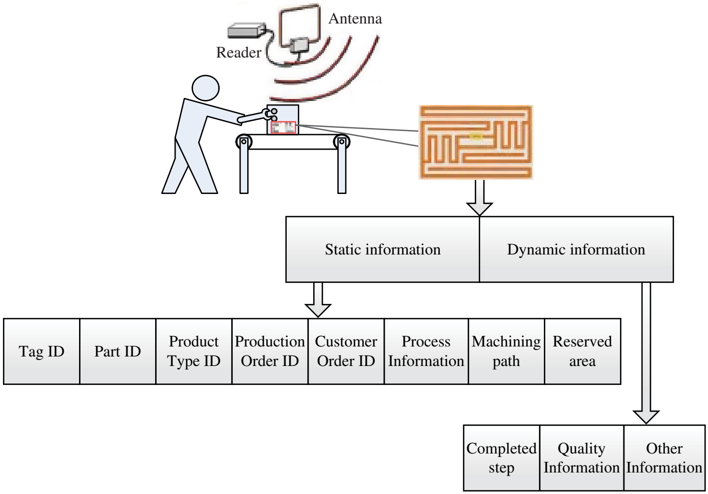

Figure 8-16 RFID tag Information.

information related to the production is stored, as shown in Figure 8-16. RFID tag information consists of two zones, the static information zone and the dynamic information zone. The static information zone is used to store static information of a material; the information in the production process is always the same, which includes the RFID tag ID, the component ID, the product type ID, the production Order ID, the customer Order ID, the process information and the processing path information. In addition, a reserved area is designed for future expansion. The dynamic information zone is used to store real-time production information, which includes information about the process and the quality of the finished parts. There are additional information areas designed for expansion demands.

In the actual implementation process, the capacity of RFID tags may be different for various reasons; a flexible implementation method is adopted in order to solve this problem. For example, basic material information is stored in the RFID tag, the process and most dynamic information can be stored in the corresponding RFID tag Agent. Although it separates information and material, it satisfies demands of the data acquisition process.

8.5 Multi-Agent RFID-Based Material Data Acquisition Systems

8.5.1 Hardware and Configuration

1) Development environment and resources
   System development platform: visual studio 2005; database: mySQL 5.0.45; System and database interface: mySQL 3.51 driver
2) RFID hardware selection
   The frequency of an RFID system is the frequency that a Reader uses to transmit data. As shown in Table 8-2, it consists of four ranges: low frequency (30 kHz–300 kHz), high frequency (3 MHz–30 MHz), UHF (300 MHz–3GHz) and microwave (2GHz–30GHz).

Table 8-2 RFID Operating Frequency Distribution.[15]

Wave Band	Low	High	Ultra High	Micro Wave
Frequency	30–300 kHz	3–30 MHz	300 MHz–3GHz	2–30GHz
Common Frequency	125–134 kHz	13.56 MHz	865–956 MHz	2.45GHz
Reading Distance	Within 0.5 m	Around 1.5 m	0.5 m–5 m	Longer than 10 m

The low-frequency RFID systems are mainly applied in short-distance, low-cost applications: for example, most access controls and fee cards. The high-frequency RFID systems are mainly applied in library management, medical logistics and other systems. UHF systems are applied in logistics, production automation, railway parcels and other areas. Microwave systems are applied in a longer reading range and high reading and writing speed occasions (narrow antenna beam direction and high prices): for example, train monitoring and highway toll systems.[14] RFID systems are also applied in supply chains, EPC Global defined two bands (13.56 MHz and 860 MHz–930 MHz) for EPC carrier frequency. For 860 MHz–930 MHz band, ISO approved the EPC

Gen 2 Class 1 UHF standard in order to make it a revision of the 18000-6 RFID air interface standard. In this paper, UHF readers are selected to manage the production process. The EPC-96I-type RFID tag is selected, which follows ISO/IEC18000-6 Type C (EPC global Class1 Generation2).

3) Reader install and setup
   This example is implemented in flexible manufacturing systems in a laboratory environment. Five RFID devices are installed in each piece of processing equipment, and the installation locations are determined by experiments. The basic parameters for each RFID device are shown in Table 8-3.
4) Encoding the RFID tag
   Due to the limited capacity of an RFID tag, the material coding portion only encodes the EPC. For the RFID tag information allocation method presented in Section 8.4.3, a simplified manner is adopted as shown in Table 8-4.

8.5.2 Material Data Process and Publish

The parameters of RFID devices are configured by using an RFID middleware Agent, devices are connected and initialized, and then the read mode is opened. The interface is shown in Figure 8-17.

Table 8-3 RFID equipment basic parameters.

ID	Data Transmission	Address
1	TCP/IP	192.168.0.111
2	TCP/IP	192.168.0.112
3	TCP/IP	192.168.0.113
4	TCP/IP	192.168.0.114
5	TCP/IP	192.168.0.115

Table 8-4 EPC-96I Coding Allocation Scheme.

		Version	Domain Management	Object Category	Serial Number
EPC-96	TYPE I	8	28	24	36

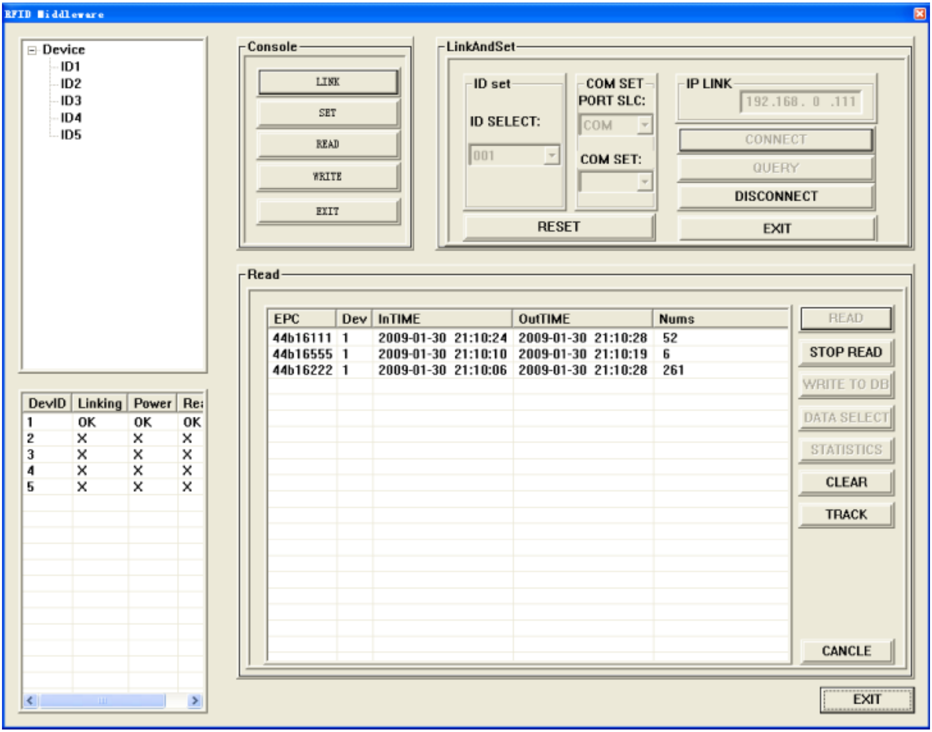

Figure 8-17 Interface of Hardware Connection and Data Acquisition in an RFID Middleware Agent.

As RFID tags enter the readable range of RFID device 1, readers get RFID tag data and upload it. The RFID middleware Agent calls API to obtain data in order to generate the original list after the event identification and treatment. The valid data list is generated by using the original data list through the filter. The required data in the valid data list are screened out by using the data selector and sent to other Agents. A valid data list is written to the database by the administrator according to the frequency or written to the database after the batch production plan is completed. The real-time data selecting and publishing interface is shown in Figure 8-18.

The refresh rate (1 s, 5 s or 15 s) is set; the frequency is the one that middleware uses to refresh identified data and process releasement; it is 1 s in this example. As shown in Figure 8-18, a valid data list is displayed in area 1, a selected result list is displayed in area 2, and the released results are displayed in area 3.

Material information can be used at any time by the upper application layer at any time through a Multi-Agent RFID-based material data acquisition system. Figure 8-19 presents a real-time dynamic material information monitoring interface. If material is in a device, then the RFID reader identifies the material RFID tag encoding. The display page also shows the material coding,; and administrators can be informed of the current position of each piece of material and its equipment running status according to the display page. If the device status displays “work” and there is not a material RFID tag at this position, then it is an abnormal event.

8.6 Conclusion

First, the RFID technology has been introduced in this chapter. Second, a Multi-Agent RFID-based material data acquisition system has been proposed in order to meet requirements of the data collection process. Third, an RFID middleware Agent, an RFID reader Agent and an RFID tag Agent have been presented in detail. Finally, the implementation process of a Multi-Agent RFID-based material data acquisition system has been addressed.

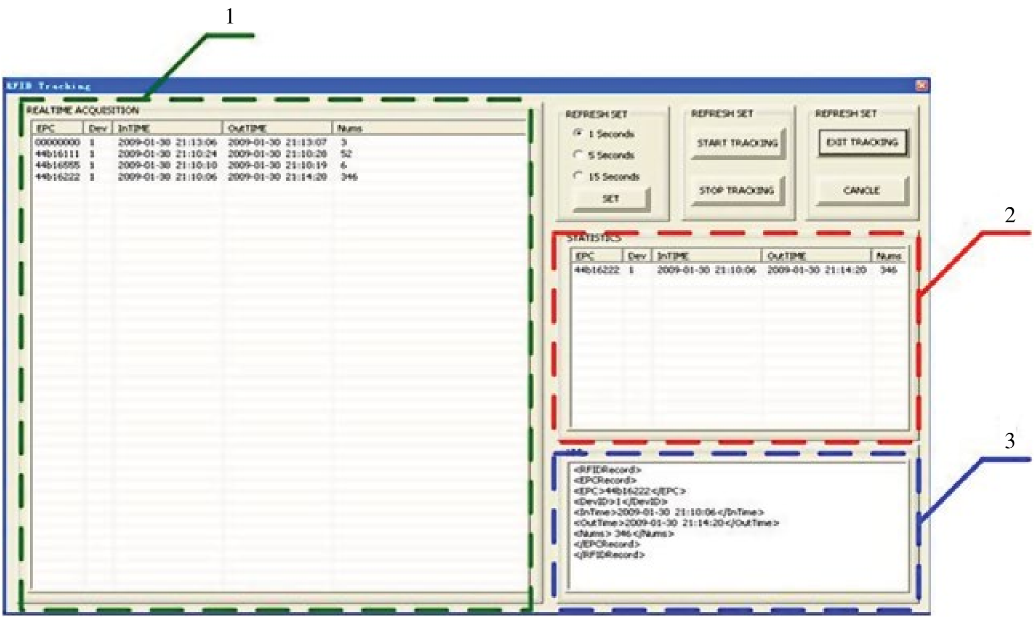

Figure 8-18 Interface of Real-Time Processing and Publishing in an RFID Middleware Agent.

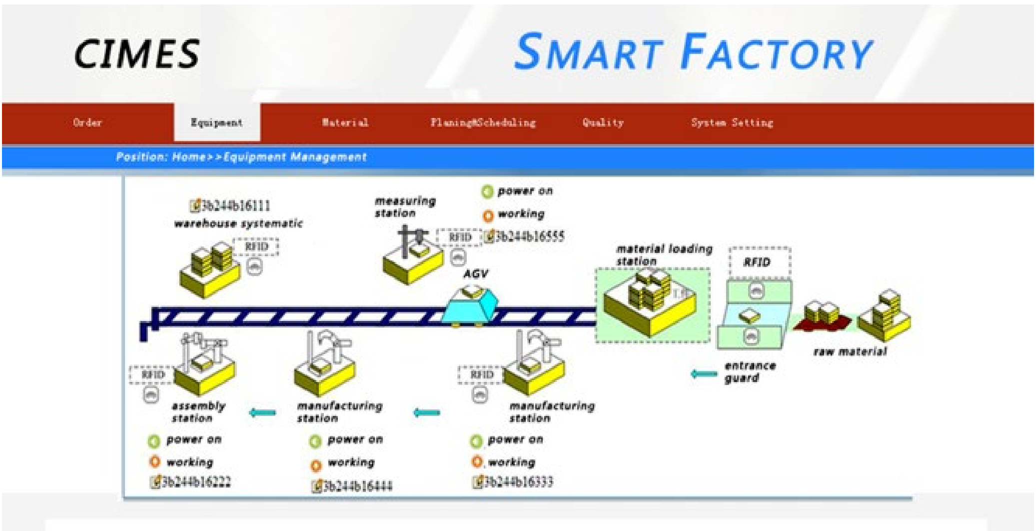

Figure 8-19 Interface of Real-Time Production Tracking.

References

[1] Chengen, W. State of the Art of Control Structures of Manufacturing System. *Control and Decision*, 1999, 14(6): 625–630.

[2] Shiyong, T. Research on RFID technology based production process tracking system. Shanghai Jiao Tong University Master thesis, 2009.

[3] Tuttle, J. R. Traditional and Emerging Technologies and Applications in the Radio Frequency Identification (RFID) Industry. 1997. *IEEE Radio Frequency Integrated Circuits Symposium, RFID, Digest of Technical Papers*: 5–8.

[4] Zhanqing, Y., Kesheng, L. *RFID Theory and Application*. Beijing: Electronic Industry Press, 2004.

[5] Chunru, X. Research and Develop on RFID based Embedded Production Data Acquisition Terminal. Guangdong Industry University Master thesis, 2006.

[6] Zhanqing, Y. *RFID Identification and Barcode Technology*. Beijing: Mechanical Industry Press. 2007.

[7] Jianyong, Y., Weimin, L., Jianqiu, Wang, et al. RFID Standard from EPC global. *Logistic Technology*, 2006, (7): 27–32.

[8] Floerkemeier, C., and Thiesse, F. EPC technology. Lecture Notes in Computer Science, Second International Conference Security in Pervasive Computing, SPC 2005: 117–118.

[9] Shimizu, M. Overview of RFID Technologies for Ubiquitous Services. IEIC Technical Report, 2004, 104(37): 49–52.

[10] Pidgeon, R. Global standard ratified for RFID. *Packaging Magazine*, 2005, 8(2):2.

[11] Weimin, L. Principle and Application of RFID technologies. Beijing: Mechanical Industry Press. 2006.

[12] Weimin, L., Zongkai, Y. Research on RFID Standard Architecture from EPC Global. *New Technology*, 2006, (3): 15–17.

[13] Zhiyu, R., Peiran, R. IoTand EPC/RFID Technology. *Forest Industry*, 2006, 1(22): 67–69.

[14] Hongzhou, H. RFID working frequency guidance and typical application. Special issue for Electronic Engineering from Taxes Instrument, 2006, 3.

[15] Matt, W. Rob, V. K. RFID: Frequency, Standards, Adoption and Innovation. JISC Technology and Standards Watch. 2006(5):10.

9

Multi-Agent-Based Equipment Data Acquisition

9.1 Introduction

A Multi-Agent material data acquisition system was presented in the previous chapter. In this chapter, the manufacturing system's equipment data acquisition technology is presented. Since the various pieces of equipment of manufacturing systems have heterogeneous software and heterogeneous network characteristics, the traditional methods have become extremely complex and are not conducive to acquiring equipment data in the current complex environment. In recent years, the equipment data acquisition technology based on OLE for Process Control (OPC) technology is gaining more and more attention. OPC technology makes it possible to communicate between a control unit and a heterogeneous manufacturing unit, to reserve a common user data interface for the upper users at the same time. This not only solves the complex data acquisition problem caused by the inconsistency of heterogeneous software and network transmission protocol, but also makes the manufacturing system management level does not have to think much about the internal implementation methods of data acquisition, which speeds up the manufacturing system manager's real-time response to manufacturing equipment, and to carry out the integration of manufacturing systems at a higher level.[1]

9.2 Basics of OPC Technology

9.2.1 Development of OPC Technology

There are many controllers and pieces of digital equipment in manufacturing systems. These devices are from a variety of manufacturers with different communication standards, and they can only form their own control systems depending on their own drivers to a certain extent, and they communicate only with specific application software. These device-dependent bottom-up solutions are mostly from the same equipment providers, which can only integrate the data from the equipment of a specific company.[2] In addition, there are third-party applications to access different equipment through the drivers provided by the manufacturer or developed by themselves.[3]

In general, since every application of the manufacturing system has its own specific requirements of data formats, protocols and other demands, which makes it impossible for a hardware manufacturer to provide drivers for all the corresponding applications, the users have to develop their own specific drivers according to their requests. This will lead to the situation that different users develop their own drivers for the same equipment. As shown in Figure 9-1, each of the three different pieces of application software should develop its own drivers for four types of devices.

The way has been shown by Figure 9-1, although the network interconnection and data integration are realized, they are not universal, and there are many shortcomings,[4] which are presented as follows:

1) Duplication of development
   Each software system developer must provide specific drivers for each particular hardware.

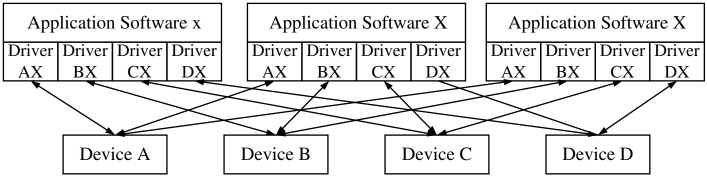

Figure 9-1 Traditional data integration methods.

2) Inconsistencies among different vendors
   According to the respective requirements of software, different data exchange protocols are adopted to develop drivers by software developers, which creates driver inconsistencies among various developers, and each driver does not support all the hardware features.
3) Changes in hardware features are not supported
   As drivers are developed by software developers, changes in the hardware features make some drivers useless. In order to adapt to new changes in the hardware characteristics, software developers must develop new hardware drivers.
4) Access Violation
   Two packages cannot simultaneously access the same device because they use different drivers. The developers tried to solve this problem by using hardware driver development methods, but the problem could not be solved due to different customers using different client agreements.

In such circumstances, better approaches should be considered for integrating information. Object Linking and Embedding (OLE) prompted for Process Control (OPC) technologies has emerged, which is a better information integration approach to solve heterogeneous devices information exchange problems.[5]

9.2.2 OPC Technology Overview

OPC technology is an industry standard.[6][7] It is a co-crystallization of the world's many leading automation and software and hardware companies, including Microsoft. This standard includes a standard set of interfaces, properties and methods, which defines the method to exchange real-time data among the client PCs by using Microsoft operating systems. This standard can be used for process control and manufacturing automation systems, and is managed by the standard OPC Foundation.[8] The OPC Foundation has more than 220 members worldwide, which contains all the major automation control systems, instrumentation and process control system companies in the world.

The OPC technology built a bridge for Windows-based applications and site process control applications. In the past, in order to record the data information of field devices, each application software developer had to develop specific interface functions. Since there are many types of field devices and products are upgraded constantly, users and software developers often have a tremendous workload. Typically this cannot meet the actual requirements of their works; it is necessary to develop an efficient, reliable, open, interoperable plug and play device driver for system integrators and developers. In this case, the OPC standard emerged.

OPC technology has been developed on the basis of Microsoft's COM/DCOM (Distributed COM) technology to enable greater interoperability among applications such as automation control systems, field devices and business offices, so as to provide a standard interface for hardware vendors and application software developers. Open and interoperable control system software is created by using DCOM technologies and OPC standards. The OPC technology uses a client/server model, in which the access interfaces are developed by hardware manufacturers or third-party manufacturers. This form solves the contradiction between software vendors and hardware vendors, completes system integration, and improves openness and interoperability of the system. OPC servers typically support two types of access interfaces, such as automation interface and custom interface, which provide access mechanisms for different programming language environments.

The automation interface is usually a standard interface defined for languages based on script; OPC client application servers can be developed by using Visual Basic, Delphi, Power Builder, and so on. However, the custom interface is specifically a standard interface formulated for C++ and other high-level programming languages. The OPC technology has become the default system interconnect solution, which brings convenience for industrial control programming, and users want to utilize the communication protocol without any worries. If one of the automated software providers does not support OPC, then it will be eliminated by history. Figure 9-2 demonstrates the data integration solution. OPC servers package the collected data from the devices into standard format, and then leave the data interfaces based on OPC. Each application software needs only one OPC client to access all the OPC servers. One server can be accessed by several clients as well.

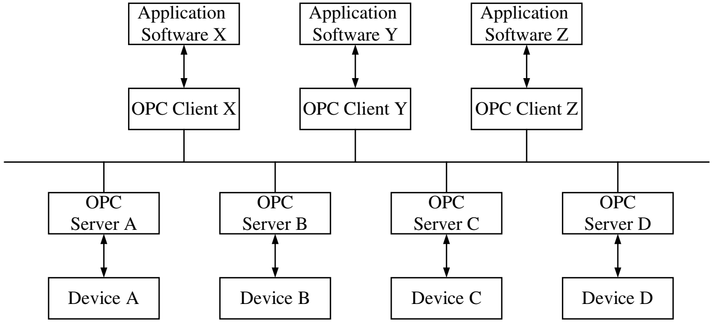

Figure 9-2 OPC-Based Data Integration Solutions.

In September 1997, the OPC foundation published OPC standard V1.0, and changed its name to Data Access standard V1.0.[9, 10] In June 2006, they published OPC Data Access standard V2.05 (OPC DA 2.05). The current version is OPC DA 3.0. In December 1998, they released Alarms & Events specification V1.0, which was upgraded to V1.02 in December 1999. Moreover, History Data Access Specification V1.0 was also released in 1998. Batch Specification V1.0 was published in January 2000. In October 2000, they released Security Specification V1.0. The specific details are presented as follows:

1) *OPC Data Access specification*. It defines a group of COM objects and interfaces in OPC servers, and formulates the criterion for clients to access the server's data. The standard consists of customizable interface specifications and automated interface specifications. This is the core of the OPC specification series and the basis of other specifications.
2) OPC *Alarms & Event specification*.[11] This specification provides a program for the servers to inform the clients about the site alarm events. It enables the industrial control software to handle a variety of alarm events in accordance with uniform standards.
3) *OPC History Data Access specification*.[12] It provides a common historical data engine that can provide additional data to interested users and clients. OPC Historical Data Access specification treats historical information as certain types of

data and uses uniform standards to integrate information of different application levels.

4) *OPC Batch specification.*[13] Based on OPC Data Access specification and IEC61512-1 international batch control standard (corresponding to U.S. standard ISA-88), it provides a way to record real-time batch data and device information. The purpose of this specification is not only to provide solutions for batch process control, but also to enable different production control schemes to work collaboratively in heterogeneous computing environments.
5) *OPC Security specification.*[14] OPC servers provide important field data. If these parameters are mistakenly changed, unpredictable consequences will follow. Therefore, it must prevent unauthorized operations. OPC Security specification provides such a special mechanism to protect sensitive data.

In the traditional implementation process as mentioned above, the OPC specifications are able to collect effectively information of underlying heterogeneous devices in the COM/DCOM technology based manufacturing systems' internal LAN. However, the COM/DCOM technology has limitations. It limits the expansion of OPC technology applications, mainly in the following areas.[15]

1) *Lack of cross-platform versatility.* Since the COM/DCOM technology depends on Microsoft platforms, the information provided by OPC is difficult to be obtained by applications based on other platforms.
2) *Difficult to integrate with Internet applications.* The DCOM technology is unable to penetrate the firewall; most DCOM-based data transmissions will be filtered out. Thus, the upper enterprise applications such as ERP manufacturing system are difficult to directly access the underlying real-time data. Meanwhile, a business OPC client cannot directly access another enterprise OPC server through the Internet.

To solve these problems, the OPC Foundation has been looking for a remote procedure call standard as a complement to the existing OPC specifications. XML languages and the appearance of Web services has filled the gap.[16] The 2002 OPC Foundation launched the XML-based XML-DA standard, which is the first step to achieve cross-platform integration and distributed applications on the Internet. In July 2003, they released OPC-XML-DA specification V1.0.

Data provided by the OPC-XML-DA specification is similar to that provided by the original COM-based OPC-DA specification. But it uses the concept of XML-based Web services, uses the Simple Object Access Protocol as a standard for packaging application-sharing messages, and uses the Web Services Description Language for describing Web services. Web services are transferred between OPC clients and Web servers via HTTP so that data communications can be successfully achieved in a variety of platforms and the Internet.

OPC-XML-DA specification supports the methods shown in Table 9.1.[17]

Table 9.1 Methods supported by the OPC-XML-DA specification.

Methods	Methods Introduction
Browse	Query name space on the server to get all the available data
GetProperties	Get information about the data items
GetStatus	Obtain the information of server, version, operation mode, operating conditions and other relevant information
Read	Obtain the value, quality and timestamp of process data
Subscribe	Order a data item list wanted by customers from the server
SubscriptionCancel	Delete the specified list of items in a previous call
SubsriptionPolledRefresh	Client asks the server to return the data entry since the last change in values
Write	Write a new value to the data item

9.3 Agent-Based Equipment Data Acquisition System

9.3.1 Requirement Analysis of Equipment Data Acquisition

Data acquisition process for manufacturing systems should satisfy the following four areas:

1) *Real-time underlying data acquisition*. A lot of real-time data should be collected by platforms. Then the underlying data should be collected, processed, uploaded and handled in the production planning and control process of the manufacturing system.
2) *Underlying heterogeneous equipment information integration*. The format of the underlying data should be unified so as to conveniently read information and give feedback when the control system is managed by a manufacturing system. Hence, it is necessary to obtain unified data information from different hardware (PLC, DCS, and so on) in different situations of a factory.
3) *Data transmission cross networks is required*. In order to integrate enterprise information and provide a convenient way for users in enterprise management platforms to understand producing situations in the equipment layer in real time, there are some problems. First of all, it is impossible to establish only one LAN, due to different location and distributed factories. If there are several LANs, synchronizing data poses a problem. Secondly, there are differences between manufacturing system environments and enterprise business environments. In a factory, industrial Ethernet networks or other forms of networks are adopted instead of Internet because devices in the manufacturing system are always based on the industrial Ethernet networks. Hence, Inter-network transmission is an objective requirement of the actual.
4) *Data integration is needed*. In order to integrate effectively the Multi-Agent production scheduling system, the Multi-Agent production control system, and the Multi-Agent production planning system, it is necessary to collect effectively equipment data, real-time send data to support real-time decisions during the production planning and control process.

9.3.2 The MAS Structure of the OPC-Based Equipment Data Acquisition

Considering the timeliness, integration and cross-platform network of data collection in manufacturing systems, this book adopts the OPC-XML-DA specification for data acquisition. The scheme is shown in Figure 9-3. The scheme consists of the underlying hardware, OPC data acquisition system, the upper application and so on. These use the OPC to collect the equipment data information including starting-ending time, waiting time, queue time, actual working hours, job status, the underlying device status of the manufacturing process and so on. The upper application sends the information to the front interface directly or to feed back to the upper management control system after information disposing. This information includes the production task status/completion, starting-ending time, resources, status, information, the actual production process and so on. The OPC data acquisition system contains the OPC server, the OPC client, the OPC XML transformation, the OPC XML-DA-DA client server and the OPC XML. The OPC client accesses the underlying device information by creating the OPC server object in the OPC server; the information transformation transforms the information into XML language; the OPC XML-DA server sends the information described by the XML language

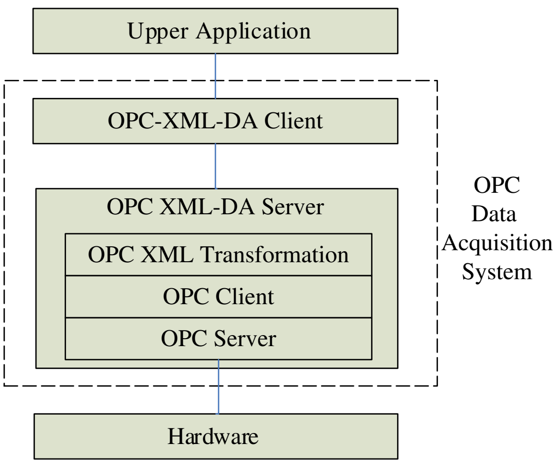

Figure 9-3 The Data Acquisition Based on the OPC Technology.

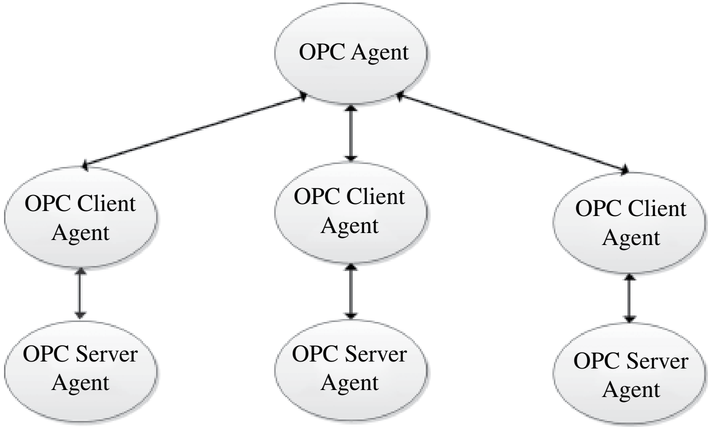

Figure 9-4 The Data Acquisition MAS Structure Based on OPC Technology.

to the OPC XML-DA client. Upper applications get the data through OPC XML-DA clients. For various manufacturing equipment, equipment providers supply the corresponding OPC server for each kind of equipment to call, many OPC servers and clients will be run at the same time in manufacturing systems.

According to the above data acquisition scheme, a Multi-Agent OPC-based equipment data acquisition system is developed to package and manage the OPC servers and clients in manufacturing systems by using the Agent technology. Its structure is shown in Figure 9-4, the system should include the following four main functions:

1) Real-time access the underlying device information.
2) Unify the obtained data in the heterogeneous environment so as to solve the data acquisition unification problem in the heterogeneous environment.
3) Solve effectively the data transmission cross-network problem. The underlying devices tend to be distributed networks, which are made up of different local area networks (LANs). We can solve problem of the data transmission across network by using equipment data acquisition.

4) Integrate the collected equipment data with other functional modules of manufacturing systems: production planning, production scheduling, production control and production process tracking, in order to upload the production data upload and formulate production orders. A unified data integration pipeline is developed to improve the efficiency of information transmission in manufacturing systems.

The Multi-Agent equipment data acquisition system contains the following Agents: OPC Agent, OPC server Agent and OPC client Agent. The OPC Agent is the coordinator in the system and manages other Agents' life cycles. The OPC client Agent acquires data from the OPC server Agent. The OPC Agent generates a corresponding OPC client Agent for each OPC server Agent in the network. The OPC server Agent gets the data directly from the hardware, and transforms it in accordance with the OPC-DA-XML specification.

In terms of the proposed structure of a Multi-Agent equipment data acquisition system, the equipment layer's information is transmitted on the Internet by using the OPC-XML-DA specification. As shown in Figure 9-5, the firewall can be divided into three layers: equipment layer, management layer and network terminal. In the equipment layer, the control equipment is different, each production unit is connected with a piece of equipment by field bus. At the same time, industrial PCs and servers are used as Internet interfaces in the manufacturing systems; many Multi-Agent data acquisition systems are adopted to be responsible for foreign communication and OPC XML data announcement. The office networks are connected with others through the Internet, which is cheaper and more convenient. Enterprises choose network servers as Internet interfaces, which communicate with servers in the equipment layer and announce web service in the Internet. In the complex and heterogeneous network situation, a Multi-Agent OPC-based equipment data acquisition system is developed to connect the manufacture system layer with the management layer. Both the Internet users and the Intranet users are available to access the underlying data in real time.

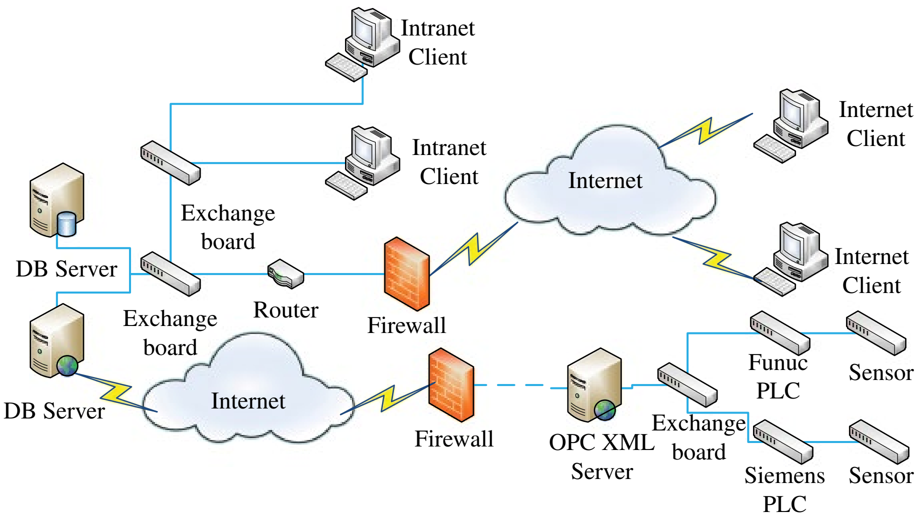

Figure 9-5 The Structure of the Manufacturing Information Integration Network.

9.3.3 The Running Model of the Equipment Data Acquisition MAS

The equipment data acquisition process is finished through the interactions of multiple Agents; the running process of a Multi-Agent equipment data acquisition system consists of the following stages: the OPC server Agent deployment and service-registered stage, the OPC client Agent creation and connection stage, and the data collection request and reply stage. In the Multi-Agent environment, there will be some exceptions such as OPC server Agent service changes, the OPC server and the client's connection failure, and so on. All these exceptions can be handled by the interactions among Agents.

The main step of the OPC server Agent deployment process is to install OPC servers and configure them so as to access their corresponding equipment. The OPC server Agent deployment stage is established after they are connected to equipment. The OPC server sends a Simple Object Access Protocol (SOAP) message to the OPC Agent to explain the supplied services (usually read and write data service) and the collected data in detail. The OPC Agent stores the information after receiving it, and returns a confirmation message. An OPC server Agent is registered at this time.

After the OPC server Agent is registered successfully, the OPC client Agent creation process is started due to the monitoring effect of the OPC Agent. The specific process is shown in Figure 9-6. The OPC Agent sends service query requests to an Agent registration center. When a new service is registered, the Agent will send the new detailed publishing service information to the OPC Agent, and the OPC Agent creates the OPC client Agent at this time. After the OPC client Agent is created successfully, the OPC client Agent will send the successful creation information to the OPC Agent. Then the OPC Agent will send the information of the corresponding OPC server Agent to the OPC client Agent. The OPC client Agent connects to the OPC server Agent according to this information, and sends feedback to the OPC Agent. Finally, the OPC Agent will save the server-client matching relationship.

After the OPC client Agent is created and connected, the data acquisition process is booting. The main task of the data

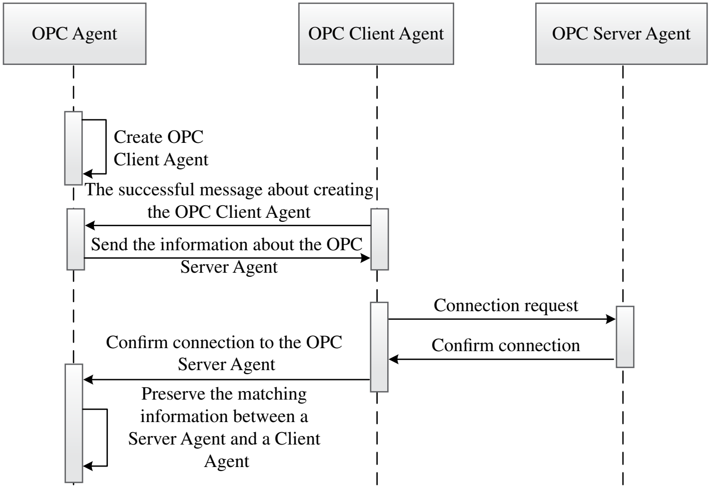

Figure 9-6 The OPC client Agent Creation.

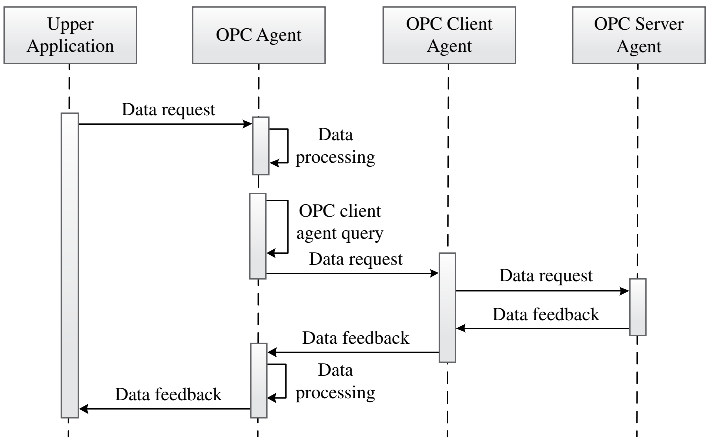

Figure 9-7 The Equipment Data Acquisition Flow.

acquisition process is to satisfy the upper application requirements of enterprises; the specific process is shown in Figure 9-7. The OPC Agent deals with the data after receiving the data request from the upper application in order to meet the OPC-XML-DA specification, and then chooses proper OPC client

Agents according to the contents of the data request, and sends data requests to the Agents. The OPC client Agent obtains the needed data from the OPC server Agent, and sends it to the OPC Agent. The processed data will be sent to the upper application in B2MML format.

9.4 Agents in the Multi-Agent OPC-Based Equipment Data Acquisition System

9.4.1 OPC Agent

As the coordinator of an equipment data acquisition system, the OPC Agent is the core of the Multi-Agent system. The upper application can obtain real-time equipment information by communicating with the OPC Agent. The OPC Agent is also used to coordinate the collaboration of all the Agents. The behavior model of an OPC Agent is shown in Figure 9-8.

The OPC Agent consists of four function modules and three interface modules. Its configuration is shown in Figure 9-9. The

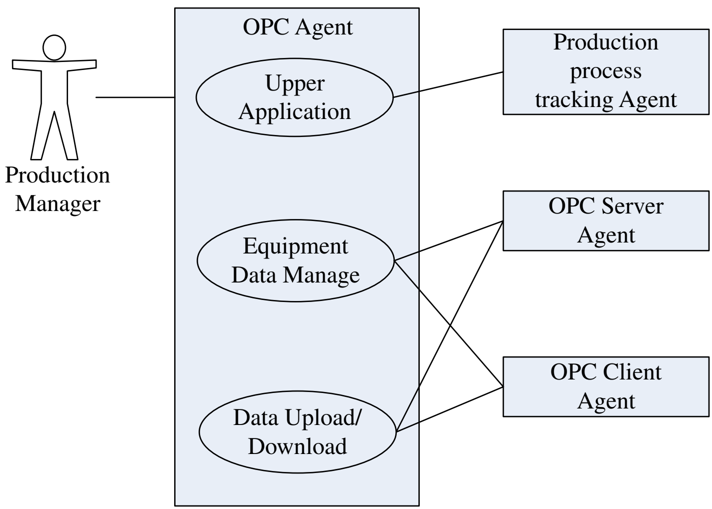

Figure 9-8 The use case diagram of an OPC Agent.

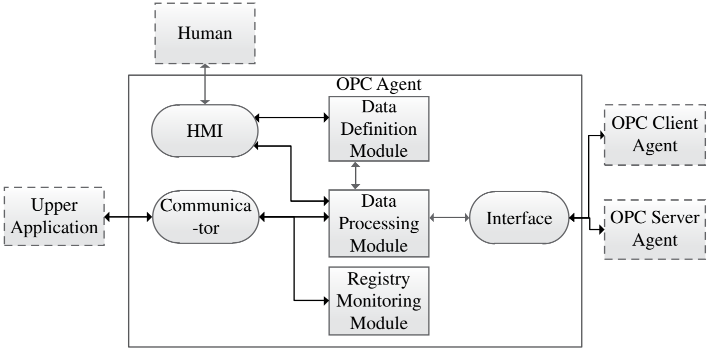

Figure 9-9 The structure of an OPC Agent.

OPC Agent has an internal data–type conversion protocol, and it needs to deal with the input and then gives outputs. The Agent belongs to the reactive Agent.

A definition module is used to define the kind and type of the information; at the same time the module is also used to configure an Agent running environment.

A registry-monitoring module is adopted to monitor the Agents' registry center in real time. When registration services are changing, the module notifies the OPC client management module to do the corresponding processing. For example, if a new OPC server Agent is added to the system, the registry monitoring module will find that the Agent center has a new server, and it will send detailed information about the server to the OPC client management module.

A data processing module converts data from the OPC client Agent to B2MML format. The module with a two-way data processing function can also transform the upper data request from a communicator into an OPC specification, which can be easily used by the OPC client Agent.

An HMI is used to provide a GUI and deal with artificial intervention and command. At the same time, HMI provides customers running status information.

An OPC interface is adopted to communicate with an OPC client/server Agent.

A communicator is used to transmit the upper data request command and real-time collected data to upper application.

At the same time, it still can obtain the Agent registry information and will store the collected data in the data pool.

9.4.2 OPC Server Agent

The first task of an OPC server Agent is to block differences among underlying devices. There are different devices running under various kinds of communication protocols, and each device connects with an OPC server. The OPC server can get the data from all kinds of PLC, DCS controllers, and can directly get the data via connected field bus sensors, actuators and other equipment; data can also be obtained from the SCADA system. Therefore, for the upper application, the underlying hardware becomes invisible. On the other hand, the underlying hardware only needs to communicate with each OPC server interface to obtain data. The behavior model of an OPC server Agent is described in Figure 9-10.

Since the OPC servers developed by providers in the industry are traditional OPC servers based on COM technology, they cannot visit across the firewall and access the Internet in real time. For this reason, an OPC XML server has been developed in this book by packing the existing OPC COM client in accordance with XML DA specification. According to its configuration as shown in Figure 9-11, the lower layer is regarded as the OPC client, which accesses and connects with each OPC data server of the underlying device through the COM/DCOM interface, and gets the data from the underlying control system. Different servers are added on the client (Server) object. The data in

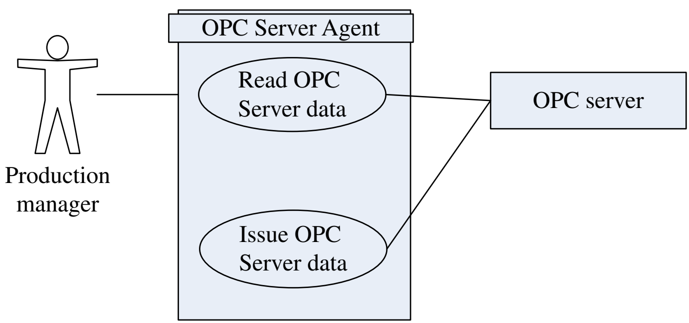

Figure 9-10 The use case diagram of an OPC server Agent.

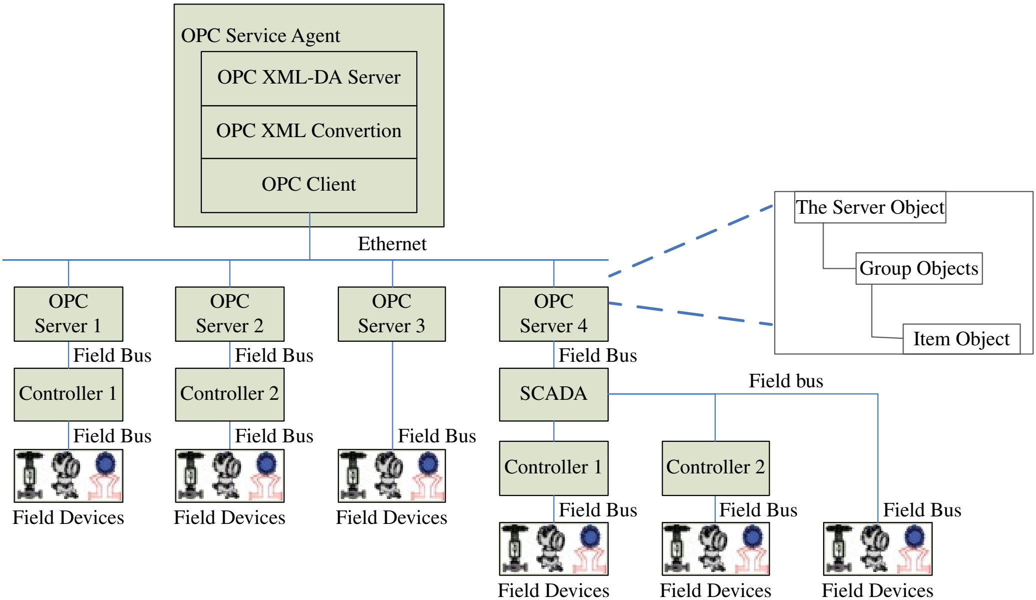

Figure 9-11 OPC server Agent structure.

different underlying hardware can be accessed at the same time by one client program, and communication in the OPC specification form is adopted to ensure real-time performance. This part provides data exchange service in the SOAP message format to the upper layer. According to the OPC data access user interface standard, OPC data accesses the automation interface, the XML specification, the SOAP protocol and the communication function requirements. As the web server, the OPC data is released and is accessed by an OPC client Agent. The heterogeneity of the underlying operating equipment is shielded by the server, and the data obtained in real time is remotely sent in XML format across the network.

The information flow of an OPC server Agent is illustrated in Figure 9-12. Its main thread is started and initialized in the

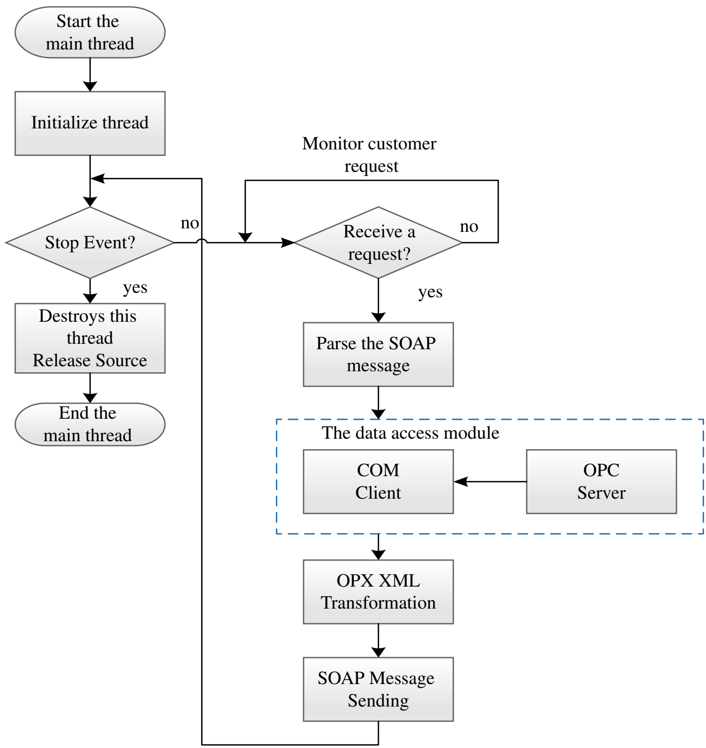

Figure 9-12 The Information flow of an OPC Server Agent.

configuration, then the state of monitoring customer requests is inserted by the system program. After the web server monitors a SOAP request from an OPC client Agent, it will parse the SOAP message and start the data access module. A COM client program obtains data from OPC servers by means of OPC automation interface servers. The data obtained from the external server are converted according to the data definition and the server type in the XML protocol and OPC interface specification, and OPC XML documents and the OPC data format are assembled, split, and analyzed in accordance with the communication so as to generate the message in the corresponding XML SOAP format and release them out to the OPC client Agent. The server operation process will be terminated when termination events occur. These events include the shutting-down-server command sent by the client and operator commands sent by the server and so on. The OPC server Agent is a reactive Agent.

9.4.3 OPC Client Agent

In Multi-Agent operating environment, an OPC Client Agent obtains data from an OPC server, and converts the data request from OPC Agent into SOAP messages according to the OPC specification. Afterwards, it converts SOAP data returned by the OPC server into the OPC format and then imports it into a data buffer. An OPC server interface is used to connect with an OPC server. The behavior model of an OPC client Agent is illustrated in Figure 9-13.

The structure of an OPC client Agent is shown in Figure 9-14.

The processing flow of an OPC Client Agent is illustrated by Figure 9-15.

After an OPC Agent is created, the OPC Agent registers in the Agent Registration Center, and then connects with an OPC Server in accordance with the address provided by an Agent. Upon connecting, the XML operation object encapsulates the data request from the Agent communication module as a SOAP request according to the OPC data structure specification, and sends it out through the OPC server

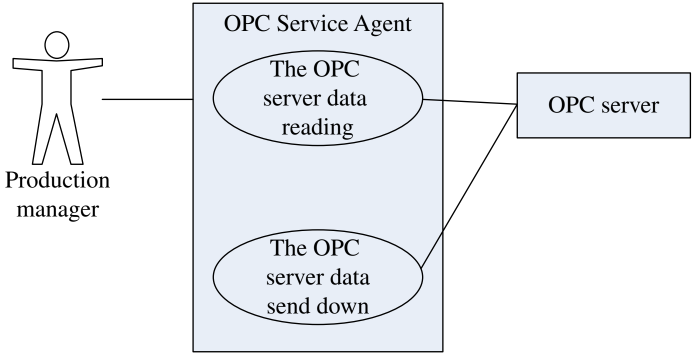

Figure 9-13 The use case diagram of an OPC Client Agent.

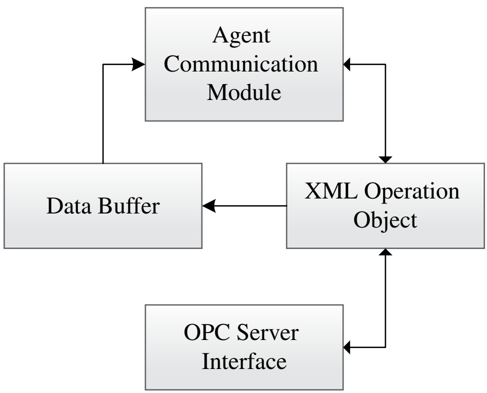

Figure 9-14 Structure of OPC client Agent.

interface. The OPC server returns the corresponding data, the XML operation object obtains data by parsing the returned SOAP message, and stores the data in the data buffer that is compliant with OPC specification. The data buffer opens the data according to the OPC specification, turns it into a structure that can be processed, and passes it to the Agent communication module.

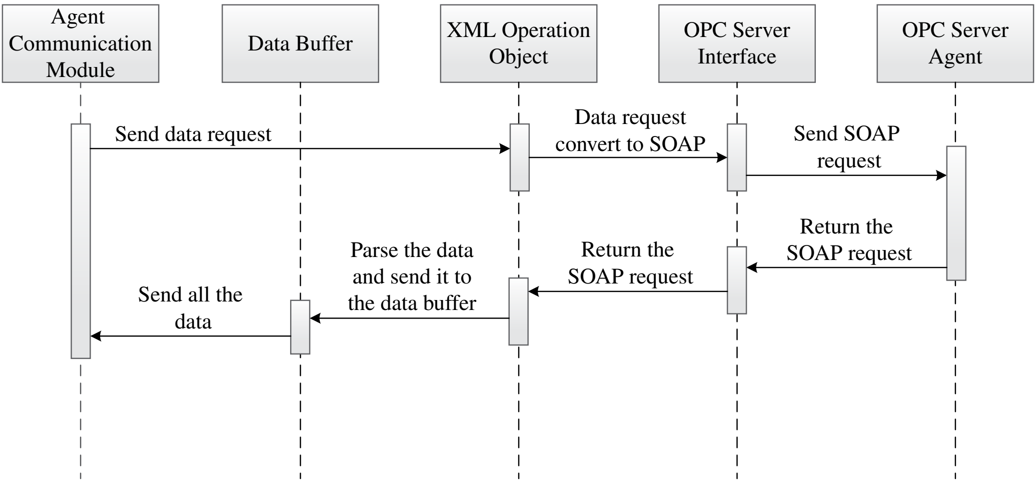

Figure 9-15 The communication timing diagram between an OPC client Agent and an OPC server Agent.

9.5 Implementation of a Multi-Agent OPC-Based System

The Programmable Logic Controller (PLC) has an important position in underlying device control problems. It is highly reliable with high resistance to electromagnetic interference. In addition, it is able to complete a variety of control functions, provides a variety of function expansion interfaces, and users can flexibly select function modules as required. In addition, the PLC programming is graphical language-based; it is simple, convenient and high flexible. Finally, PLC generally has low power consumption, smaller size and weight. Thus PLC is widely applied in the industrial field, becoming a master device of CNC machines, transport trolley, robots, conveyors, and so on. Nowadays many manufacturers provide PLC globally and have tried to promote their own communication protocols. Therefore, this brings a great deal of inconvenience for selecting internal control equipment, collecting data and integrating data in manufacturing systems.

Currently many PLCs have networking capabilities, which support multiple field bus protocols. However, the controller interoperability, interaction between the control layer and the upper layer application have become complex due to non-uniform data definitions and formats. This will cause high cost, and the open problem of the system cannot be solved. Therefore, a Multi-Agent OPC-based equipment data acquisition system has developed and applied in PLCs.

9.5.1 System Hardware and System Network Architecture

9.5.1.1 Network Protocol

The emergence of the fieldbus has played a huge role for the equipment of the automatic system; however, the field bus has high cost, low speed and limited support applications. There are many different kinds of field bus, and it is very difficult to integrate them. Ethernet has advantages such as low cost, high transmission rate, rich resources of software and hardware, good openness and compatibility that has been widely employed. The high-speed Ethernet overcomes its defect, and becomes an industrial Ethernet in the industrial fields. The compatible industrial Ethernet with common Ethernet in technology (e.g., the IEEE802.3 and IEEE802.3 U standard) can be used as a coordinative level and unit level network. Hence, the industrial Ethernet is chosen as a connection of PLC to PLC and PLC to network of computer controlling.

9.5.1.2 Hardware

Siemens S7 200 PLC and CP243-1 have been chosen as the controller hardware and the Ethernet module, respectively. The CP243-1 was developed by Siemens which uses it for the expanding function module through connecting the S7-200 PLC system to the industrial Ethernet. CP243-1 can use STEP 7 Micro/Win 32 on S7-200 remote configuration, programming, and diagnosis. It can undertake communication with other similar PLC and communicate with the server based on OPC as well. So, the CP243-1 has been chosen as a PLC network interface.

9.5.1.3 System Network Architecture

System network architecture adopts a layer structure, and the machine layer connects to the upper computer through the industrial Ethernet. Remote users can access the system through the Internet. The system hardware network architecture is illustrated in Figure 9-16.

CP243-1 for Ethernet network has a "server/client" configuration: a particular PLC in the network is the server, other PLCs and PCs are the clients, and each client can be connected to only one server. In this system, all the equipment information interacts with the AGV, so it is configured as a server, and other equipment are clients. As long as the IP address of the server is set in the client program, the network interconnection can be realized as indicated in Figure 9-17.

9.5.2 Data Integration Based on OPC Technology

The Siemens S7-200 specially provided for PC Access is selected as the OPC server, which provides data to the OPC client and can communicate with any standard OPC client. PC Access software comes with the OPC client test side; the user can easily detect and configure communications correctness of their projects.

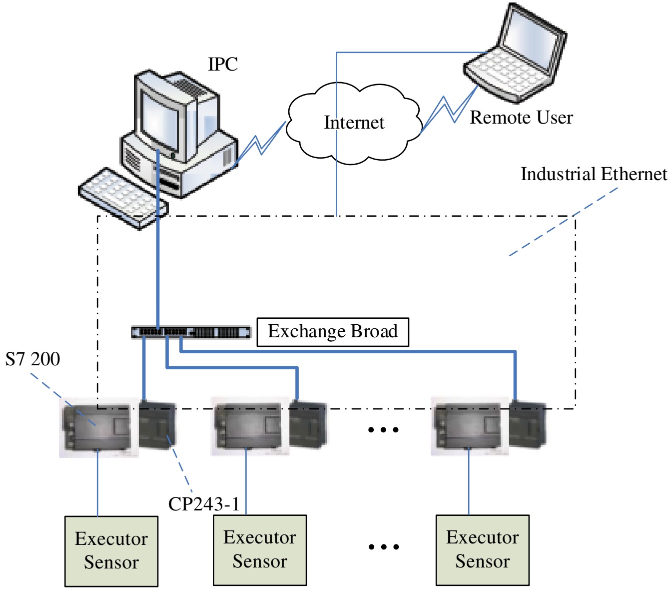

Figure 9-16 System hardware network architecture.

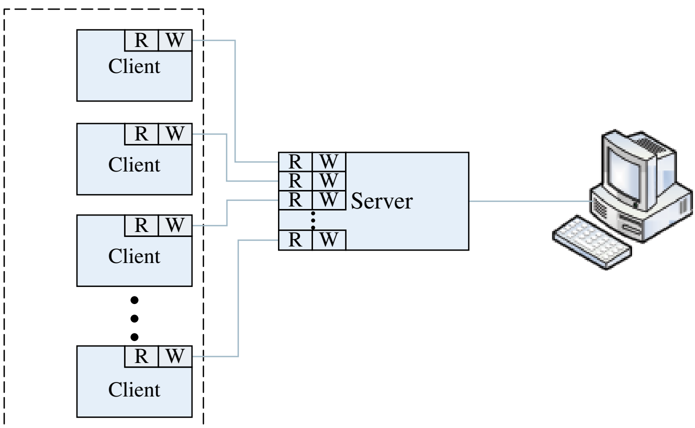

Figure 9-17 System network configuration.

PC Access can be used to connect Siemens, or third-party PC software supporting OPC technology, which supports the OPC Data Access version 3.0; it can run on Windows 2000 or Windows XP. The symbol table can be imported from the Micro/WIN project, which supports the new S7-200 smart cable (RS-232 or USB). The following describes how PC Access achieves information integration.

As already mentioned, during the network configuration, the IPC (that is PC Access) is configured as a client. So as long as the IP address of the Server PLC is set in PC Access, the communication between PC Access and PLC can be achieved. In this case, data entry should be added in the PC Access (Figure 9-18); then it will be able to access the PLC data.

Since remote users hope to monitor real-time information on the PCL via the Internet, as the data acquisition module in the upper information system, PC Access itself does not have the ability to release XML; users need an OPC COM client in the remote to read PC Access first. Subsequently, these data are consolidated to a structure (*item*, *value*, *quality*, *timestamp*) via a simplified OPC XML server, and encapsulated in

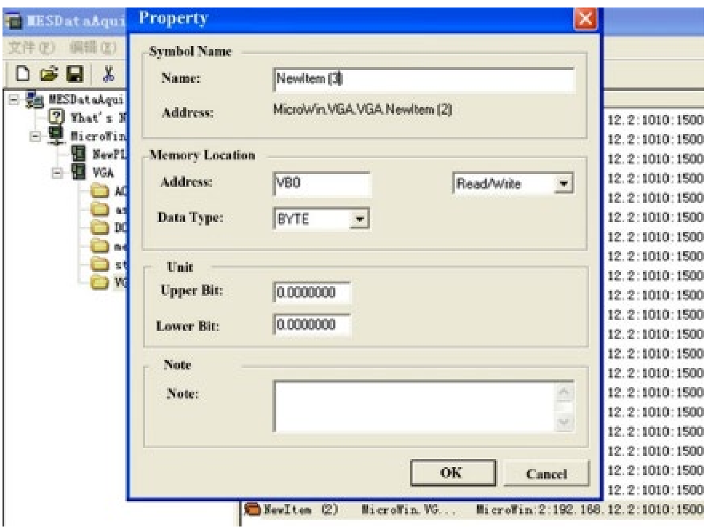

Figure 9-18 Add data to PC Access.

Item ⊂ *Group* ⊂ *Server*. The OPC client gets information in the OPC XML server through the subscribed server entry in order to ensure timeliness in real time. Because different servers are simultaneously loaded, the underlying isomerism cannot be perceived by the upper layer, and the information can be easily integrated in the general OPC client. Binary data is converted into XML format on the OPC XML server for publishing, which is done by the OPC client Agent. The information process of data acquisition is shown in Figure 9-19.

The most important function of an OPC Agent is to provide users data and the configuration window of an OPC Server. As shown in Figure 9-20, there are five areas on the screen, each

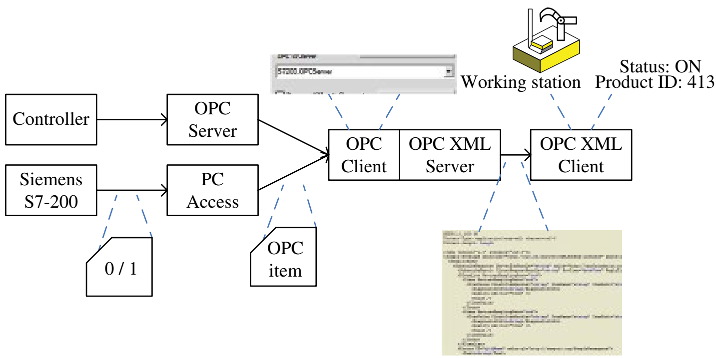

Figure 9-19 Information process of data acquisition

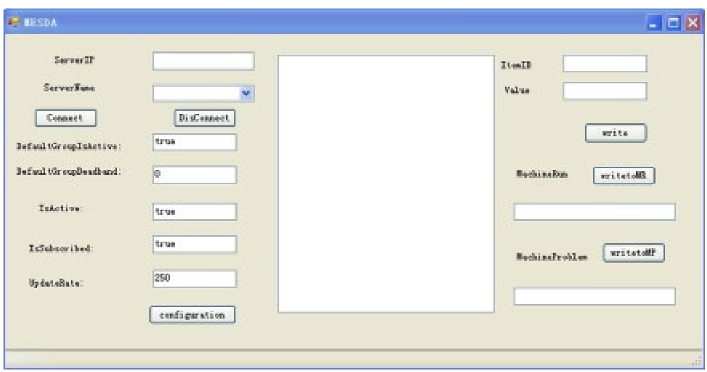

Figure 9-20 OPC Agent configuration interface.

Figure 9-21 Real-time data acquisition in the remote devices.

region in the platform interface and their functions are presented as follows:

1) *Establish OPC Server Agent object area*. Enter target OPC server Agent's IP address in the corresponding text box of Server IP, and then enter the name of the OPC server Agent in ServerName. Click "Connect" or "Disconnect" to establish a connection.
2) *Set group properties area*. Input group property values in each corresponding text box, click "configuration" to configure the properties of a group.
3) *Select Item Area*. Select the desired Item in the Checklist Box.
4) *Write Item value*. Input Item ID and values to the Item; click "write" to do the writing operation on a particular Item.
5) *Write into the database*. Click "writetoMR" to collect data. Then the collected data is written to the database.

Through the above configuration, an OPC Agent can access controller information remotely. Figure 9-21 shows the device status information of the device controller obtained in real time, the green color displays the working status of the controller.

9.6 Conclusion

In this chapter, OPC technology for equipment data acquisition has been presented. Then the data acquisition requirement has been analyzed in the heterogeneous environment. Subsequently, a Multi-Agent OPC-based equipment data acquisition system has been developed. The OPC Agent, the OPC server Agent, and the OPC client Agent have been designed as well. The implementation process of a Multi-Agent equipment data acquisition system has been introduced at the end of this chapter.

References

[1] Zhang, G. Research and Implementation of Key Technologies on Flexible Manufacturing System. Shanghai Jiao Tong University, 2009.

[2] Reinhard, F. A Component-Based Application Framework for Manufacturing Execution Systems in C# and.NET. *ACM International Conference Proceeding Series Archive*, 2002: 169–178.

[3] Jim, L. OPC Brings XML-DA to the Factory Floor. *Control Solutions*, 2002, 7: 34–38.
[4] Zhu, Y. The Development and Research of OPC Data Access Server. North China Electric Power University, 2003.
[5] Zhang, C., Sheng, Z., and Guo, G. An Approach to Integration of Producing and Management Information Based on OPC. *Computer Engineering*, 2002, 3: 217–220.
[6] Yang, J. and Chen, X.. A New TechnoIogy of Process Control—OPC. *Measurement-Control Technology*, 1999, 18(11): 56–57.
[7] OPC Foundation. OLE for process control data access standard version 2.0. 1998.
[8] http://www.OPCfoundation.org/.
[9] OPC Foundation. Data Access Custom Interface Standard Version 2.05 A. June 28, 2002.
[10] OPC Foundation. Data Access Automation Interface Standard Version 2.02. February 4, 1999.
[11] OPC Foundation. OPC Alarms and Events Version l.02. November 2, 1999.
[12] OPC Foundation. Historical Data Access Custom Interface Standard Version 1.1. January 26, 2001.
[13] OPC Foundation. OPC Batch Custom Interface Specification Version 2. July 19, 2001.
[14] OPC Foundation. OPC Security Custom Interface Version 1.0. October 17, 2000.
[15] Peng,J. and Wang, J. Newly Released OPC Specification—OPC XML-DA. *Metallurgical Industry Automation*, 2004 (1): 6–9
[16] Fussell, E. Opening Standards in enterprise integration. *In Tech*, 2003, 50 (12): 21–25.
[17] OPC Foundation, OPC-XML-DA 1.00 specification. http:www.opcfoundation.org. July 12, 2003.

10

The Prototype of a Multi-Agent-Based Production Planning and Control System

10.1 Introduction

The previous chapters introduced Agent-based production planning methods, production scheduling methods, and production control methods for complex manufacturing systems, such as Job Shop manufacturing systems and Re-entrant manufacturing systems. In order to integrate Agent technology with RFID technology and OPC technology, the prior chapters also developed a Multi-Agent material data acquisition system and a Multi-Agent equipment data acquisition system to collect material and equipment data in the manufacturing process. To apply and validate these technologies and methods, this chapter shows how the prototype of an Agent-based production planning system and control is designed and properly implemented.

10.2 Architecture of a Prototype System

10.2.1 The Software Architecture

The prototype of an Agent-based production planning and control system is a logical architecture that is composed of a human-computer interaction process, a running kernel and an operating environment on the basis of a network. The human-computer interaction process is adopted to manage information exchanges with users. The running kernel manages and controls the control flows, data flows and information flows in order to maintain the operation of the system. The operating environment concerns

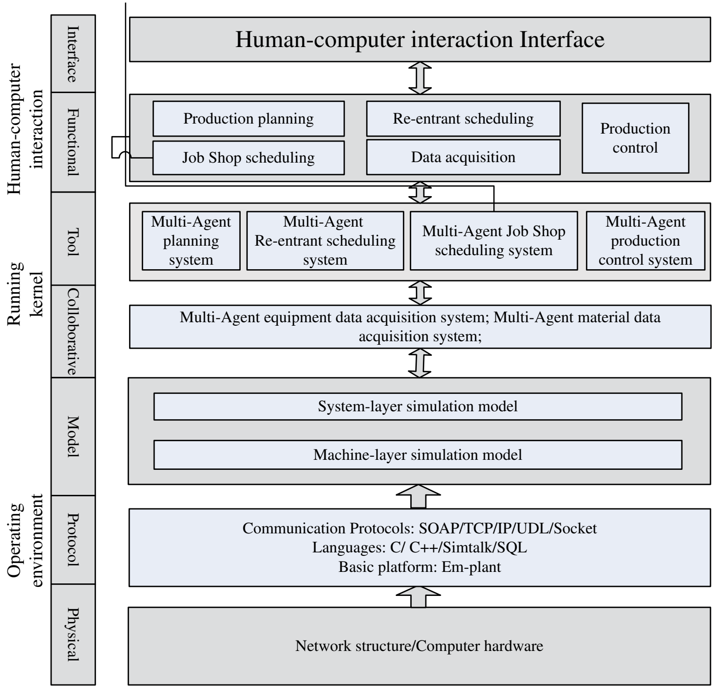

Figure 10-1 The software architecture of the prototype system.

the hardware environment of a system. Such an architecture is based on the logical architecture of the prototype of production scheduling and production control manufacturing systems. Relevant software platforms can be classified as follows: the interface layer, the functional layer, the tool layer, the collaborative layer, the model layer, the physical layer and the protocol layer as shown in Figure 10-1.

1) Interface layer
   Interfaces contain the production planning interface, the production scheduling interface for Job Shop manufacturing systems, the production scheduling interface for re-entrant manufacturing systems, the production control interface, equipment data acquisition interface and the material data acquisition interface. The interaction process between human and computer can be achieved through the above interfaces.

2) Functional layer
   Through collaborations of various functional models, the system simulates the operation process of a manufacturing system, and validates production planning methods, production scheduling methods and production control methods. The functional modules include the production planning module, the production scheduling module for Job Shop manufacturing systems, the production scheduling module for re-entrant manufacturing systems, the production control module, the material data acquisition module based on the RFID technology and the equipment data acquisition module based on the OPC technology. The data acquisition module is able to reconfigure the data acquisition interface according to the status of material and equipment.
3) Collaborative layer and tool layer
   Being the operation environment of a Multi-Agent production planning and control system, the collaborative layer and the tool layer construct the core of the collaborative process in the prototype system through the Agent. They are similar to the distributed operating systems with special designs, where cross-computer platforms, operating systems and network systems provide functional modules with necessary services, including data management, statement management, object model management, ownership management, time management, and data release management. Data release management and time management are complex, mostly affect the system, and are difficult to implement.
4) Model layer
   The model layer is used to simulate resources in manufacturing systems in order to reproduce the production process in Job Shop manufacturing systems and re-entrant manufacturing systems. The simulation model library is composed of simulation models of a machine layer, a machine group layer (a cell layer), and a manufacturing system layer, in order to enable the hierarchical modeling.
5) Physical layer and protocol layer
   As the bus of the platform, the distributed object technology based on SOAP is employed by the protocol layer. It is

composed of three parts: 1) the SOAP package policy, which defines a framework to describe the message content in detail; 2) the SOAP response representation, which defines the protocol for remote procedure calling and response; 3) the SOAP encoding rules, which define a serialization mechanism to exchange data instances defined in application programs. The protocol runs on the basis of TCP/IP protocol. Due to the interface restriction of some applications, for instance, eM-Plant provides socket communication interface; the platform also supports Socket communications.

10.2.2 The Hardware Architecture

The hardware architecture of Agent-based production planning and control prototype system are presented in Figure 10-2. In the architecture, the prototype systems are connected by Ethernet with the hardware in manufacturing systems. The relational database, the production planning and control system, and the data collection system are connected through the switches. The architecture thus collects data in real time, stores data, further supports the production planning and control process based on real-time data. Among common methods that construct industrial control systems, such as Ethernet, field bus and serial connections, this book employs an industrial Ethernet to connect devices and PCs in the laboratory, and adopts RFID technology and OPC technology to collect the real-time data of equipment and materials.

10.3 Agent Packages and Communication in a Prototype System

Abstract components constituting the prototype system and their connections have been described in the architecture, which demonstrates communications among components in detail. In the implementation stage, these abstract components can be further classified as actual Agent and MAS in order to implement specific functions.

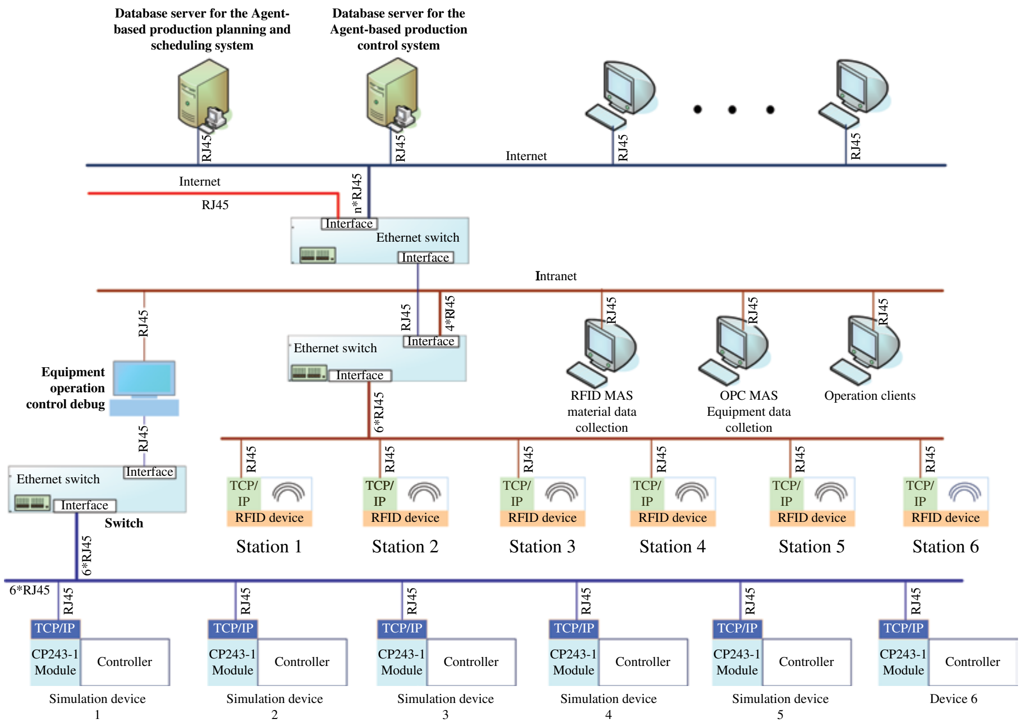

Figure 10-2 The hardware architecture of the prototype system.

10.3.1 The Agent Package Method

In a Multi-Agent system, the Agent is closely related to distributed objects involved in the production planning and control process. The distributed object technology provides basic tools for organizations and packages of a Multi-Agent system. There are two specifications for distributed objects in the world: CORBA and COM/DCOM. Both specifications provide the underlying support for software component technologies and apply technologies produced upon them, that is, OpenDoc and OLE/ActiveX. To implement the "plug and play" function of an Agent, the standard object interface should be defined on the basis of the Agent interface definition language – KQML. Through language mapping in KQML, users can write the interactive Web Service between servers and clients via NET, and interact information among Agents via the SOA technology.

To exchange information among Agents, KQML interfaces should be defined at the first step. KQML interface descriptions used in object implementations and client calls are independent from operation systems and programming languages. According to the information exchanged between objects, the object's interface is designed and described by KQML language. Abnormal operations are defined so as to make the system robust and friendly. The process for writing interfaces is illustrated in the following "Task bidding" process for a collaborative scheduling Agent.

ModuleAgent
Interface TaskBid//Bidding interface

```
{
Excedtion Reject{longErrorNum};
void announceToLeaguer(outstring Task_
no,outstring Part_id,outstring Item,
outlong Parteenumber,outstring Due_
time,outstring Statetime,outstring Reply_
with):
void bidFromLeaguer(instringTask_
no,instringPart_id,infloatBid_value,instring
Finishjime, instringStatetime, instringRe-
ply_with, instring Reply_with);
```

```
void awardTOLeaguer (outstring ask_
no,outstring Part_id,Reply_with_to):
};
interfaceaskPlanning//accessing request to
mission planning
{
ExceptionReject{longErrorNum}:
Void Preselected BidList(outstring Bid_
Path,outstring Label_1);
Void winnerList(instring Plan_path,instring
Label_2);
Outstring;
}://endmoduleAgent_SC
```

where parameters in the function *announceToLeaguer* represent the bidding task information, that is, "task number", "sub-task number", "task description" "quantity" "earliest starting time", "completion time", and so on. For unsuccessful bidding, the system might rebid some tasks in the form of asynchronously datagram calls without return, where the parameter *Replywith* represents the message sign of a bid.

Parameters of the function *BidFromLeaguer* represent the bidding information of Agents of collaborative members, which are similar to parameters in the bidding tasks. Parameters *Reply_with_to* corresponds to *Reply_with* in bidding messages. Function *preseletedBidList* describes the request information of a collaborative Agent to a demand management Agent. Since a task Agent belongs to the information Server Agent, which belongs to the collaborative scheduling Agent, only information of ability control and decision-making are passed among them, and data information is directly obtained from corresponding databases. The parameter *Bid_Path* describes path names of databases to store information of all the bidding tasks. Parameter Label transfers a number to information receivers to read data correctly. For example, "0" represents reading from the first line in the database. Function *winnerList* represents the Agent list with bidding winners in the task scheduling process of the task Agent. The collaborative scheduling Agent returns evaluation results of bidding information, whose parameters are similar to the function *presectedBidList*.

Several KQML files are generated based on definitions of KQML interfaces. These files are translated to become interface communication programs used in applications in clients and in servers through the KQML compiler. To implement the applications, client programs are written to stimulate and handle operation requests in the defined objects. Moreover, codes that can respond to requests in clients are also required in servers. The client interface programs are included in projects of client application programs to generate executable client programs, and the server programs are included in projects of SOA servers to generate executable server programs. The Server object class is registered in the implementation database through the object adapter. In this way, servers can operate automatically when clients use the Server object. When designing application programs in clients, users need to consider choosing whether statistic calls or dynamic calls to send requests. Compared with dynamic calls, statistic calls have advantages such as simpler programs, better performance and more reliable type checking, but they are not flexible. When employing dynamic scheduling, KQML files must be added in the interface library.

10.3.2 The Communication Implementation Model of Agents

The communication model of Agents adopts a hierarchical structure, as shown in Figure 10-3. The internal module of Agents in the top layer is composed of the knowledge database of Agents, the logical reasoning mechanism of Agents and so on. The cooperation and negotiation between Agents are solved in the protocol and interpreter module. The explanation and description module based on XML implements proposition, description and explanation of message contents. KQML provides Agents with unified languages and acts. Based on package technologies and service technologies provided by SOAP, the message receiving and sending module communicates and interacts between Agents.

In terms of this structure, a unified communication interface can be defined for each Agent, which contains data and operations provided by Agents. Here is a basic framework for communication interfaces:

Interface Communication

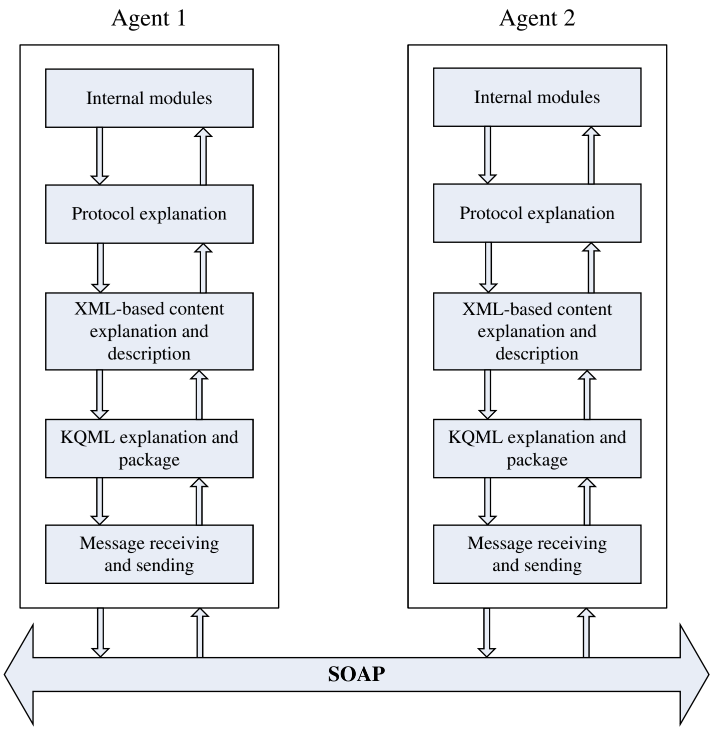

Figure 10-3 The communication model for Agents.

```
{
  void    SendMessage(out String description);
//Send messages;
  void    ReceiveMessage(in String descrip-
tion);  //Receive messages from related
Agents;
  BOOL  DataOut(out file* data);     //
Respond to related Agents
  BOOL  DataIn(in file* data);       //
Receive response messages from related
Agents
  String   SeekPartner(outMessagenumber);
//Seek cooperation
  String   StopCooperate(outMessagenumber);
//Stop cooperation
};
```

where *file* and *Message* are predefined data structures to define forms of information and messages sent by Agents.

10.3.3 The Message Classification of Agents

Functions of a production planning and control system are implemented through communication and collaboration among Agents, where communications can be classified as information delivery and service requests. The information delivery is that sender Agents proactively deliver the relevant information to recipient Agents. The service request is that sender Agents ask recipient Agents for necessary services and further return service results to the recipient Agents.

Message delivery among Agents and MASs in the prototype system are illustrated in Figure 10-4. Through continuous message sending and service response among Agents and MASs, the functions of production planning and control system are realized.[1]

P represents a Multi-Agent production planning system. CA represents an order/product demand management Agent. S represents a Multi-Agent production scheduling system. C represents a Multi-Agent production control system. R represents a Multi-Agent data acquisition system. Collaborative relationships between Agents in Figure 10-4 are explained as follows:

1) Information exchange between CA and P.
   CA-P01: bidding information of orders.
   P-CA01: tendering information of orders.
   P-CA02: awarding information of orders.
   CA-P02: tracking information of orders.
2) Information exchange between P and M.
   P-M01: material information of products required in orders.
   P-M01: resource information required in orders.
3) Information exchange between P and R.
   P-R01: material preparation plans.
   R-P01: material preparation reports.
   P-R02: material preparation processing advice.
   P-R03: resource utilization inquiries.
   R-P02: resource utilization reports.
   R-P03: resource purchase plans/equipment maintenance plans.
   R-P04: material and resource purchase plans.

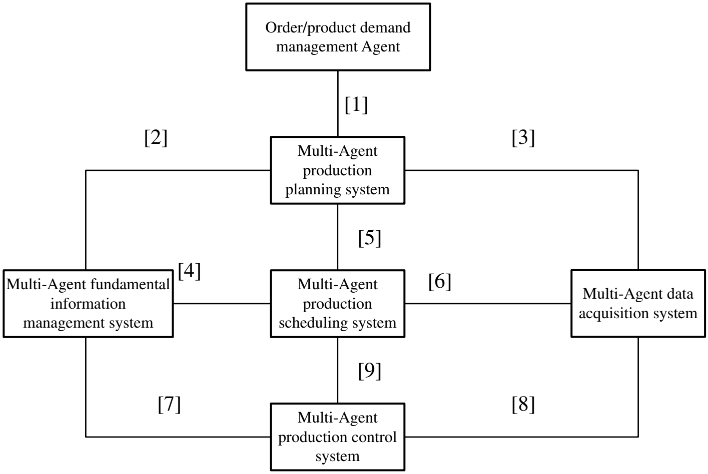

Figure 10-4 The collaborative relationships between Agents in production planning and control.

4) Information exchange between S and M.
   S-M01: information requests for product structure.
   S-M02: process information requests for product components.
   M-S01: releasing product structure information.
   M-S02: releasing process information of product components.
5) Information exchange between P and S.
   P-S01: plan assignment.
   S-P01: plan completion notice.
   P-S02: plan change notice.
   S-P02: dynamic information notice in production process.
6) Information exchange between S and R.
   S-R01: resource plans.
   R-S01: resource plan arrangement.
7) Information exchange between M and C.
   C-M01: task lists of products.
   M-C01: equipment and material list required in products.
   M-C02: process list.
8) Information exchange between R and C.
   C-R01: resource request/equipment request/maintenance request.
   R-C01: resource allocation plans/equipment maintenance plans.
   R-C02: resource utilization inquiries.
   C-R02: resource utilization information.

9) Information exchange between S and C.
   S-C01: job order arrivals.
   S-C02: task modifications/temporary task arrivals.
   S-C03: task fulfillment inquiries.
   S-C04: task bidding information.
   S-C05: starting time of tasks.
   S-C06: temporary task notice.
   C-S01: task fulfillment notice.
   C-S02: task delay owing to machine breakdown.
   C-S03: tendering information.
   C-S04: tasks allocated to machines

Therefore, Multi-Agent production planning, scheduling and control process plays a leading role in the production planning and control system, and other Agents and MASs are in subordinate positions that respond to its requests and provide services. This is a collaborative relationship based on master-slave services. For example, there are master-slave services in production planning MAS and data collection MAS. In addition, there is another peer collaborative relationship between MASs and Agents, and MASs and Agents are on equal positions to provide services for each other. For example, the relationship between the production planning Agent and the production scheduling Agent is peer.

10.3.4 Realization of the Communication Mechanism of Agents

In a production planning and control system, the messages between Agents are composed of header, primitive, content and number.[2] The header consists of two parts: the recipient name and the sender name. Such a message structure is not suitable for broadcast messages. Therefore, messages should be repeatedly sent to other Agents for messages to be broadcast. If the message is a response to another message, the number here refers to the number of the responded message. The number contains information about the message delivery time.

The system mainly uses primitives in KQML as follows:

"Ask": ask for other Agents for services or inquiries
"Announce": mainly for task biddings

"Tell": inform other Agents about related message
"Broadcast": broadcast messages
"Bid": bidding information
"Award": confirm messages of bidding
"error": indicate wrong formats of messages
"sorry": indicate recipient unable to answer for requests

The system also adds two primitives: *Deliver-Release*. The former passes data about orders and jobs, the latter releases production commands that must be performed by recipients.

Message example:

```
Tell
::content ( COrder<<01<<A<<Hongqi Machinery
Factory<<2001/3/16<<3000<< )
::receiver taskAgent
::language Order
```

This message is sent from the order/demand management Agent to the collaborative planning Agent to inform the recipient about details of an order. The recipient calls upon the order handling methods to deal with the process content of the message to generate a new order object. It also gives message data to the corresponding attribute. This process includes converting strings into numbers, time and converse handlings. Since numerous data are required to describe an order in detail, mainly the characteristic information or index information are included in the message delivery. Detailed or additional process data and product data are dynamically read from the Multi-Agent fundamental data management system if required by Agents.

10.4 The Manufacturing System Simulation in a Prototype System

With respect to disadvantages such as high operation costs and difficult maintenance in real manufacturing systems, simulation techniques are adopted in the prototype system. The Agent-based production planning and control prototype system is examined by constructing a simulation platform for manufacturing systems.

10.4.1 The Manufacturing System Simulation

Simulation techniques are important means and methods to analyze movements, explain dynamics and investigate the rules of systems.[3] They developed very fast after the first computer was invented in the 1940s. Especially in recent years, with the development of control theories, computing technologies and information processing technologies, as well as computer hardware and software, system simulation techniques have achieved many breakthroughs. Remarkable achievements have been made in theoretical research, industrial applications, simulation engineering, and tool development environments. Simulation techniques have already become an independent comprehensive science. They are the methods that investigate objects based on similarity principles and analogy relationships. Simulation techniques investigate objects indirectly based on models that have similar features and variations.[4]

The basic procedure of simulation is shown in Figure 10-5. The key steps are summarized as follows:

1) *Problem abstraction:* analyze a system to generate a target system;
2) *Simulation:* generate simulation data by executing the simulation model;
3) *Data collection:* collect data from the real system;

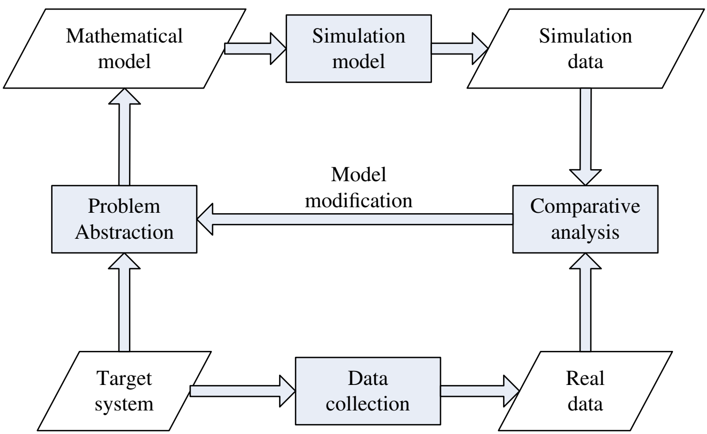

Figure 10-5 The basic principle of simulation.

4) *Comparative analysis:* compare simulation data with real data in order to evaluate fitness of the model;
5) *Model modification:* revise the simulation model on the basis of data analysis results in order to make the simulation model become as close to the target system as possible.

With the continuous development of network technologies, the modern simulation environment has become increasingly complex. Simulation systems are mostly multi-platform, multi-system complex applications, which require current simulation application solutions to become compatible to support different system platforms, data formats and multiple connectivity options. It is necessary to achieve loose coupling, cross-platform, language-independent, reliable distributed simulation in the network environment. Higher interoperability and reusability are required in distributed simulation. Currently, a large number of researchers closely follow the development direction of modeling and simulation, employ network technologies to extend traditional simulation technologies and develop theories of distributed simulation and parallel simulation. The characteristics of the distributed simulation include distribution, interaction, heterogeneity, time-space consistency and openness.[5] In a distributed simulation system, each simulation node is distributed in physical locations, as well as in functions and computing power. They can run interactively through networks or run each simulation independently.

A simulation model of manufacturing systems is developed based on simulation technologies; this simulation model communicates and exchanges information with an Agent-based production planning and control system. Compared with traditional simulation models, the simulation model has following requirements:

1) Part of the simulation model is associated with Agents.
2) Simulation objects can interact with Agents to respond to changes in manufacturing systems, and have a feedback feature.
3) The simulation-based manufacturing systems collaborate with MASs, which are able to investigate dynamic characteristics of manufacturing systems. They are able to obtain specific response measures of manufacturing systems in dynamic environments by the interaction among Agents at the macro level and the micro level.

Meanwhile, with respect to the distributed nature of manufacturing systems, distributed computing is necessary for manufacturing system simulation. The research of distributed computing is a widely investigated field of computer science. There are various problems in distributed systems from theory to practice, which include distributed programming languages, distributed computing theories, operating systems, communication, reliability, data management and application and so on. These are presented in detail in relevant literatures and thus are not discussed here. In terms of distributed simulation and its requirement analysis, the simulation process should satisfy the following requirements:

1) *Clock synchronization and update consistency.* Since the event delay caused by calculation and communication in simulation is different from the real delay, the clock synchronization mechanism is applied to ensure consistency between orders of simulation events and real events in order to guarantee causal accuracy in simulation.
2) *Distributed object service.* Distributed object service organizes simulation resources in the network according to objects; a clear access interface is defined for each object. The middleware shields the underlying computing development such as interoperability between objects, network hardware, and protocol heterogeneity in order to ensure that simulation application developments can focus on the simulation layer and the model layer.
3) *Message transmission and data distribution management.* With respect to communication redundancy and excessive overhead in distributed simulation, message transmission and data distribution management are used to forward the data between simulation applications according to data supply and demand relationships between simulation entities. Data are also filtered to ease the pressure of redundant data on system resources.
4) *Load balance.* Distributed simulation decomposes the system into multiple logical processes that communicate with each other. As the resource management module of a distributed simulation system, the load distribution redistributes simulation loads among processors reasonably and transparently to optimize the comprehensive performance of a system.

5) *State data storage and fault tolerance.* State data storage and fault tolerance can be used to recover unexpected errors in the simulation process, to obtain the same initial conditions for testing optional methods, and to realize continuous execution and exception handling in long term simulation.

The production planning and control prototype system is verified through building simulation models, production planning, scheduling, control activities are developed on the basis of simulaand analyzing performances of manufacturing systems. These processes are implemented with the following requirements:

1) The simulation model of a manufacturing system is built according to requirements of a production planning and control system.
2) Since the simulation models and the production planning and control system are developed in different languages and environments, systems are integrated in heterogeneous environments.
3) The interfaces between simulation models, the production planning, and control system with configurable functions, the production planning and control system can configure the data interface according to the number of devices in simulation models in order to transmit data.
4) Interoperability between the simulation process and the production planning and control system is required to ensure data interaction between the planning and control system and the simulation system.

The discrete event simulation tool eM-plant and SimTalk language are employed to program the model of an Re-entrant manufacturing system shown in Figure 10-6. Hierarchical modeling methods are applied to Re-entrant manufacturing systems, the hierarchical model is composed of a system layer, a machine layer and a machine group layer. eM-plant is useful to provide equipment utilization and system yield in manufacturing systems efficiently.

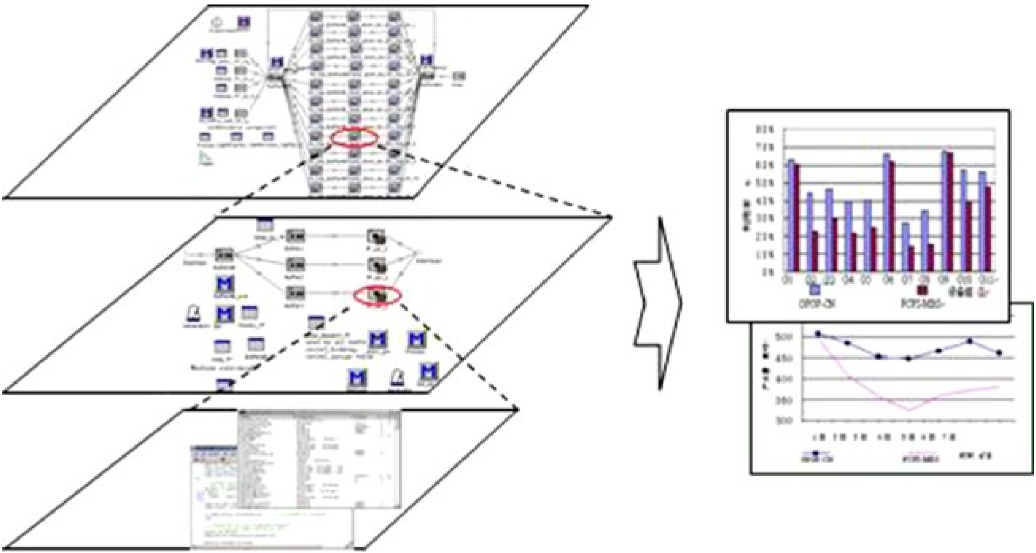

Figure 10-6 A simulation modeling instance of Re-entrant manufacturing systems.

10.4.2 The Information Interaction Logic Architecture between the Prototype System and the Simulation Model

The common distributed mechanism at every part of the system is considered in the logic architecture. In the prototype system, the common mechanism mainly refers to the information exchange mechanism between simulation models and Agent software systems. The interactions between traditional simulation tools and other software systems are achieved by using API to call other software systems based on the Remote Procedure Call Protocol (RPC). RPC usually calls remote services via COM and DCOM interfaces. Its disadvantages are complex programming processes, limited service access according to specified operation systems, and difficult promotion on the Internet. As a sequential information exchange mechanism, SOAP can be used to exchange data type instances defined by applications, which is simple and easy. Consequently, a remote access server method based on SOAP is developed in this book to implement the information exchange mechanism.

The distributed elements in every part of the system are also considered in the logical architecture, which mainly refers to data generation, data control and data management. These elements form a unitary structure through data exchange. With respect to the practicality of the SOAP technology in integrating data, the logical architecture based on SOAP technology in the prototype system is adopted in this book, as shown in Figure 10-7, where the production planning and control management process are supported by information exchanges in the prototype system.

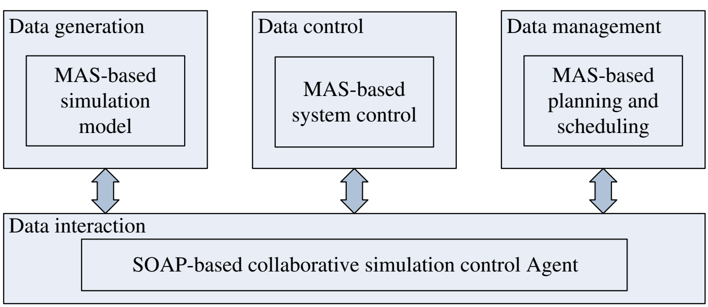

Figure 10-7 The logic architecture of a manufacturing control system.

The SOAP-based information exchange mechanism links up data generation, data control and data management, as presented in Figure 10-8, which mainly contains a service provider, a service user, a Web Service server and a local machine. According to the information exchange requirements, these three elements can be regarded as service providers and service consumers, respectively. The interaction process between a Web Service server and a local machine is presented as follows:

1) The service provider determines the name and content of services that it can provide, and expresses them in WSDL language through the Web Service server. And then the content of described services is standardized by UDDI standardization integration.
2) Service users ask for solutions in the UDDI components of a Web Service server according to required services. A UDDI server analyzes the contents of service requests from service users and generates the path to services.
3) The service user's local machine sends service requests to the Web Service based on the SOAP protocol. The Web Service

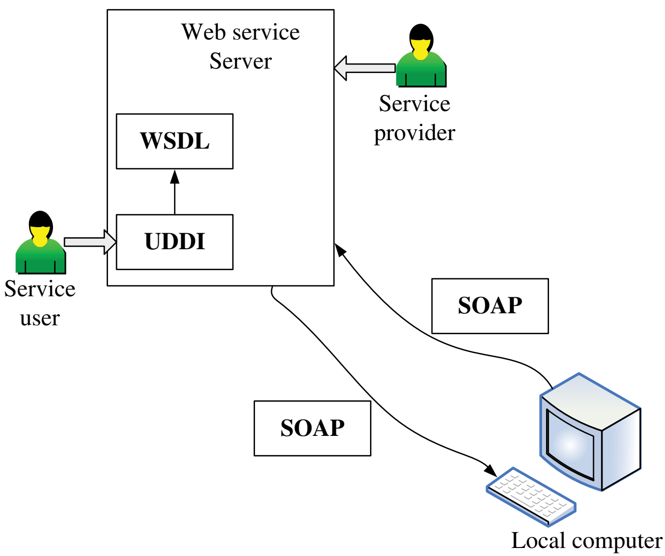

Figure 10-8 The SOAP-based interaction model for Web service.

performs services analysis after receiving requests from service users and sends the specific service to the service user's local machine based on SOAP protocol.

10.5 Software Implementation and Application of a Prototype System

10.5.1 Function Design of a Prototype System

In terms of requirements of the Agent-based production planning and control process in manufacturing systems, functions of a prototype system are designed, as depicted in Figure 10-9. The prototype system contains the following functional modules:

1) Production planning for distributed manufacturing systems
   - *Order management:* collect customer orders, convert order demands to product tasks, return completion of orders.
   - *Production planning:* analyze workloads of critical bottleneck resources in the planning period, determine the type and quantity of final products required in the planning cycle, determine the order delivery time.
   - *Production plans inquiry:* provide production plan inquiring, modifying, deleting and production plan completion inquiring in distributed manufacturing systems.
2) Production scheduling for Job Shop manufacturing systems
   - *Positive feedback scheduling:* allocate tasks to resources in Job Shop systems while considering process constraints, resource capacity constraints and other conditions of tasks, assigning resources and completion times to product tasks, generate positive feedback to optimize the operation plan.
   - *Negative feedback rescheduling:* reschedule periodically, based on negative feedback, and then generate new operating plans in order to effectively manage various types of dynamic events in Job Shop systems through real-time access to relevant data.
   - *Operation plan inquiry:* provide production plan inquiring, modifying, deleting and production plan completion inquiring in Job Shop manufacturing systems.

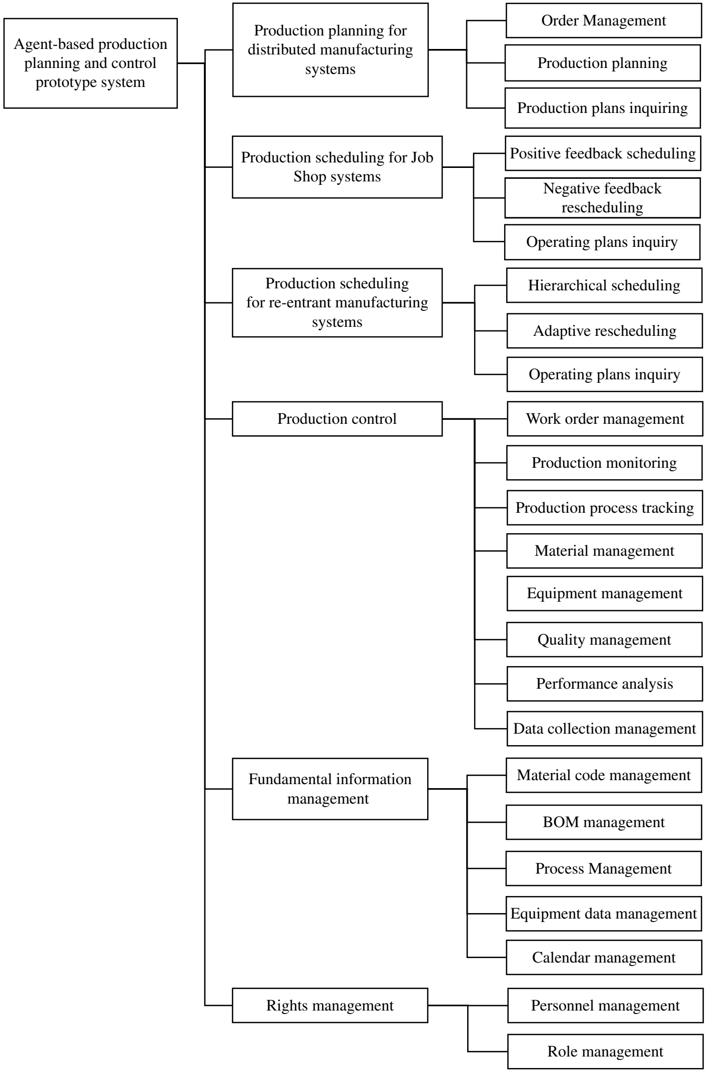

Figure 10-9 Basic functional modules.

3) Production scheduling for Re-entrant manufacturing systems
   - *Hierarchical production scheduling:* allocate resources and completion times to product tasks based on process constraints, resource capacity constraints and other constraints in the system layer and the machine layer of re-entrant manufacturing systems and generate operation plans.
   - *Adaptive rescheduling:* develop the assessment model for dynamic events in re-entrant manufacturing systems, determine the rescheduling layer and generate new operation plans based on adaptive rescheduling decisions by using fuzzy logic methods.
   - *Operation plans inquiry:* provide production plan inquiring, modifying, deleting and production plan completion inquiring in re-entrant manufacturing systems.
4) Production control
   - *Work order management:* generate work orders for production equipment, personnel and other resources within a certain time or shift based on operation plans.
   - *Production monitoring:* visualize collected data, monitor tasks being performed, respond to implementation results of operation tasks and emergencies in operations, provide timely feedback to the planning layer and the scheduling layer once exceptions happen.
   - *Production process tracking:* collate and classify collected information in the production process, which includes production tracking activities, material tracking activities and quality tracking activities, in order to meet the requirements of the production process tracking.
   - *Material management:* manage movement, buffer and storage of materials and provide data basis for material data tracking.
   - *Production performance analysis:* provide the latest evaluation report of the actual manufacturing process.
   - *Equipment management:* track and guide maintenance activities for machines and tools.
   - *Quality management:* record and analyze quality information in the manufacturing process to control quality.
   - *Data collection management:* manage hardware and software interfaces related to RFID and OPC, ensure the stability of the real-time data collection process.

5) Fundamental data management
   - *Material code management:* manage material and provide each material with a unique identification code.
   - *BOM management:* define technical files for product structures to guide the generation processes of production plans and operating plans.
   - *Process management:* describe processing steps and operation sequences for products.
   - *Equipment data management:* manage equipment and tools and provide equipment with unique identification codes.
   - *Calendar management:* manage the factory calendar, which includes sabbatical mode, working hours, overtime and other settings.
6) Rights management
   - *Personnel management:* manage system operators.
   - *Role management:* manage different roles and define their permissions.

10.5.2 The Running Process of a Prototype System

The running process of an Agent-based production planning and control prototype system consists of the following modules: Multi-Agent production planning system, Multi-Agent production scheduling system, Multi-Agent production control system, Multi-Agent material data acquisition system, and Multi-Agent equipment data acquisition system, and so on. The running process of a prototype system is shown in Figure 10-10.

1) *production planning, scheduling and control:* accept customer orders via the Agent-based production planning and control system, publish production plans, operation plans, work orders and production control commands via collaboration among a Multi-Agent production planning system, a Multi-Agent production scheduling system, a Multi-Agent production control system, a Multi-Agent fundamental information management system and a Multi-Agent data acquisition system, simulate production processes by using simulation models.
2) *Data acquisition in running processes:* collect equipment information and material information via a Multi-Agent

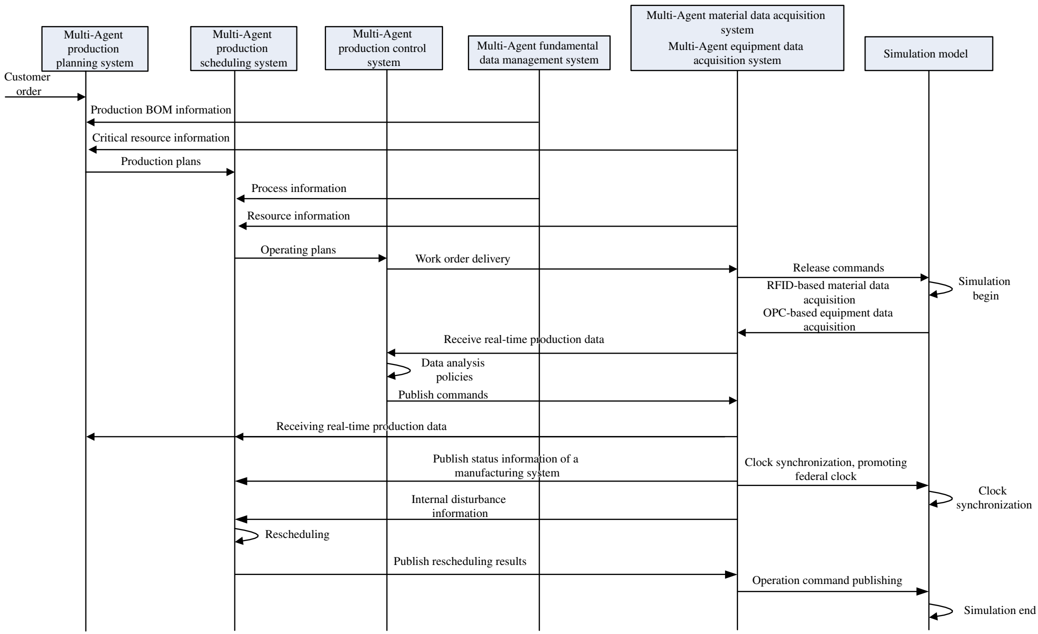

Figure 10-10 The running process of a prototype system.

equipment data acquisition system and a material data acquisition system, which is the basis for production planning and control activities in running processes.
3) *Rescheduling policies under disturbance:* simulate the production planning and control process under internal or external disturbances such as rush orders. For example, the Multi-Agent production scheduling system publishes rescheduling policies and makes decisions when machines break down, and then regenerates production control commands to ensure the stability of manufacturing systems.
4) *Event simulation until completing all events:* obtain production information from a Multi-Agent equipment data acquisition system and a Multi-Agent material data acquisition system. Compared with production plans, operation plans and work orders in the Agent-based production planning and control system, statistically analyze production processes.

10.5.3 Production Planning in Distributed Manufacturing Systems

As regards distributed manufacturing systems, information among multiple factories of a distributed manufacturing system is completely shared or incompletely shared, a Multi-Agent production planning system employs a contract net-based collaborative protocol and an auction-based negotiation protocol to solve production planning problems. Production managers can arrange production plans reasonably by using the production planning module; the running process is shown in Figure 10-11.

1) *Production planning parameter maintenance:* maintain the data related to the critical resource table, the process table and BOM involved in the production planning process of distributed manufacturing systems. In the test, product demand information is published by the order/demand management Agent according to customer demands.
2) *Architecture parameter configuration in distributed manufacturing systems:* configure architecture parameters in distributed manufacturing systems due to the difference between optimization models and optimization policies

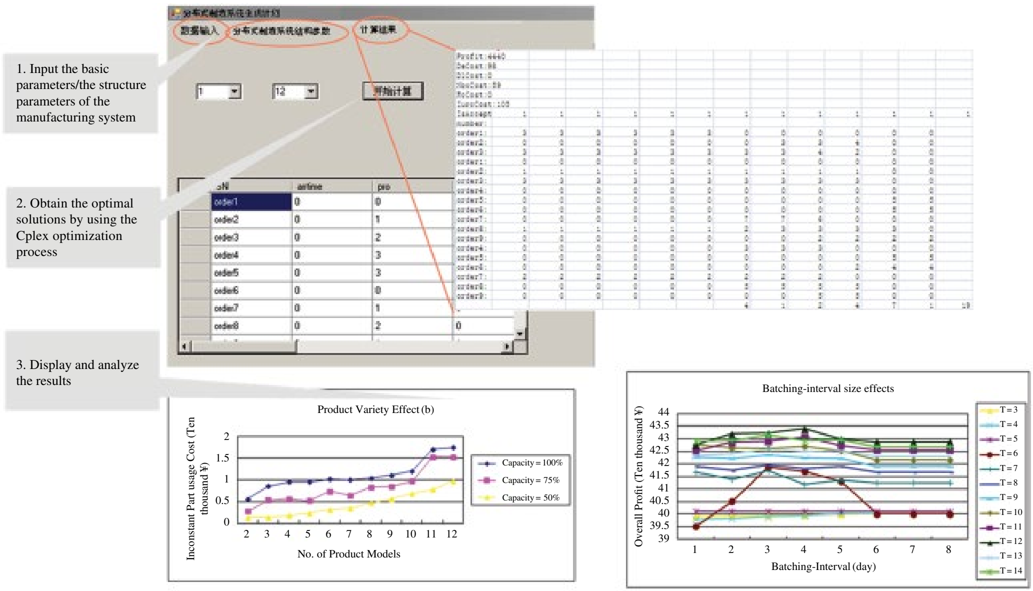

Figure 10-11 Production planning in the Multi-Agent production planning system based on a collaborative protocol and a negotiation protocol.

while considering completely shared information or incompletely shared information in manufacturing systems.

3) *Production planning:* generate production plans for distributed manufacturing systems by using the contract net-based collaborative protocol and the auction-based negotiation protocol. The mathematical programming tool (e.g., Cplex) is employed to generate solutions for the underlying system in the optimization process.
4) *Results analyses of production plans:* analyze the optimization performance of production plans in manufacturing systems with different organizational structures and different length of planning periods.

10.5.4 Production Scheduling in Job Shop Manufacturing Systems

As regards production scheduling problems in Job Shop systems, the complexity of the problem and the dynamic manufacturing system environment are taken into consideration. A Multi-Agent production scheduling system is developed, and then the scheduling method based on positive feedback strategies and the rescheduling method based on negative feedback are designed. They optimize operating plans and maintain stability of manufacturing systems during operation processes, whose procedure is presented as shown in Figure 10-12.

1) *Production plan acceptation:* accept production plans of distributed manufacturing systems and generate the task list in Job Shop manufacturing systems; the task list contains material number, name, quantity, delivery time and so on.
2) *Scheduling based on the positive feedback strategy:* maintain the data related to the critical resources table, the process table and BOM involved in production planning process in Job Shop manufacturing systems. In the test, parameters in the hierarchical genetic algorithm are set to optimize production scheduling results.
3) *Analyses of production scheduling results:* obtain the Gantt chart of Job Shop manufacturing systems and analyze machine utilization.

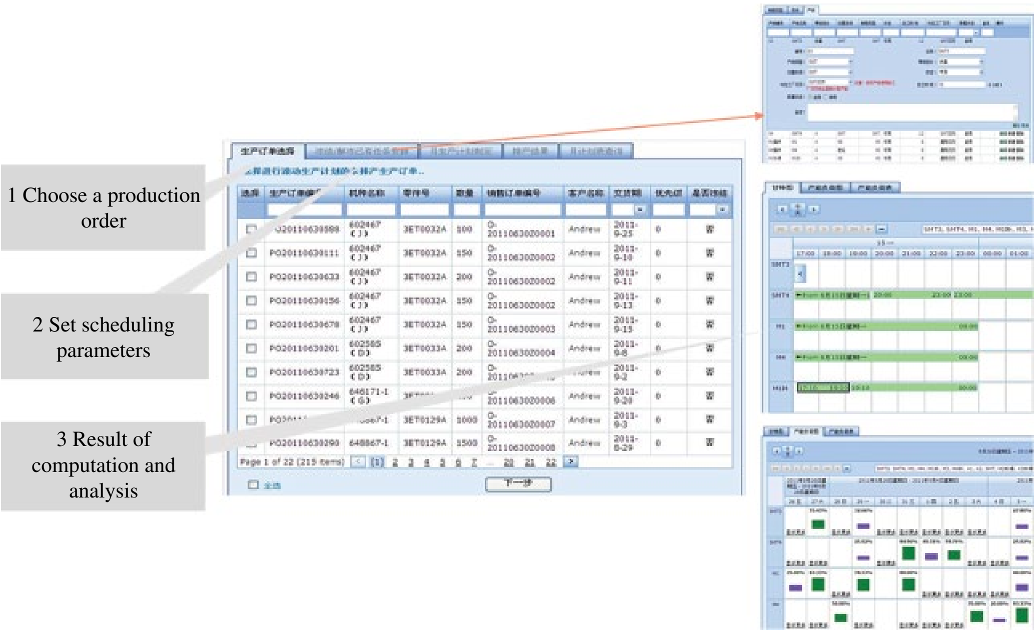

Figure 10-12 The positive feedback production scheduling process in Job Shop manufacturing systems.

In the actual manufacturing environment, sudden changes such as rush orders and random machine breakdown are taken into consideration in the rescheduling process. In this case, the rescheduling method based on the negative feedback strategy is proposed for the rescheduling problem with rush orders. The rescheduling results are published by the Multi-Agent production scheduling system. The changed operating plans after rescheduling, are optimized to ensure the stability of production processes in Job Shop manufacturing systems, as shown in Figure 10-13.

10.5.5 Production Scheduling in Re-Entrant Manufacturing Systems

As regards Re-entrant manufacturing systems, a hierarchical adaptive production scheduling method is developed in this book due to their particular organization. At the system layer, a global collaborative scheduling method based on a combinatorial auction is proposed to solve the global production scheduling problem by collaboration amongst a set of Agents. The running process is shown in Figure 10-14.

1) *Resource data maintenance:* maintain the data related to machine resource tables, process tables and WIP tables. In the test, data of ordered WIP tables are provided by the task Agent.
2) *Material parameters setting:* determine the priorities of different re-entrant products while considering the delivery time of orders.
3) *Feeding plan input:* generate the feeding plan based on upper production plans.
4) *Production scheduling period setting:* determine the production scheduling period according to the complexity of real production processes.
5) *Production scheduling methods at the system layer:* generate the global production scheduling plan in reentrant manufacturing systems based on the combinatorial auction algorithm.
6) *System-layer global rescheduling:* perform the system-layer global rescheduling process when the system is greatly disturbed. (i.e., the feasibility of operating plans is affected.)

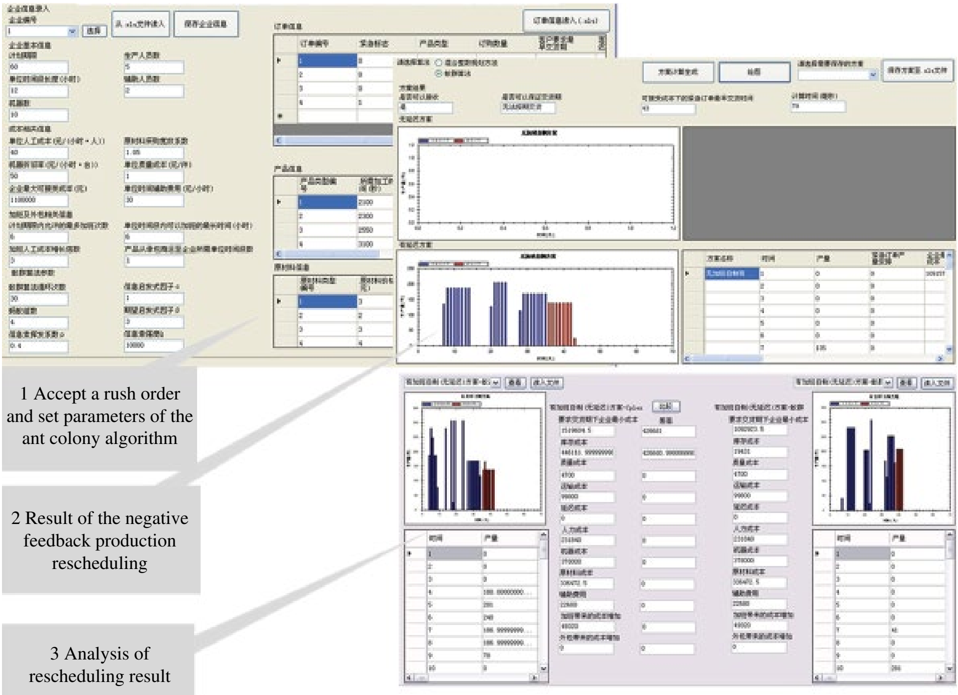

Figure 10-13 The negative feedback production rescheduling process in Job Shop manufacturing systems.

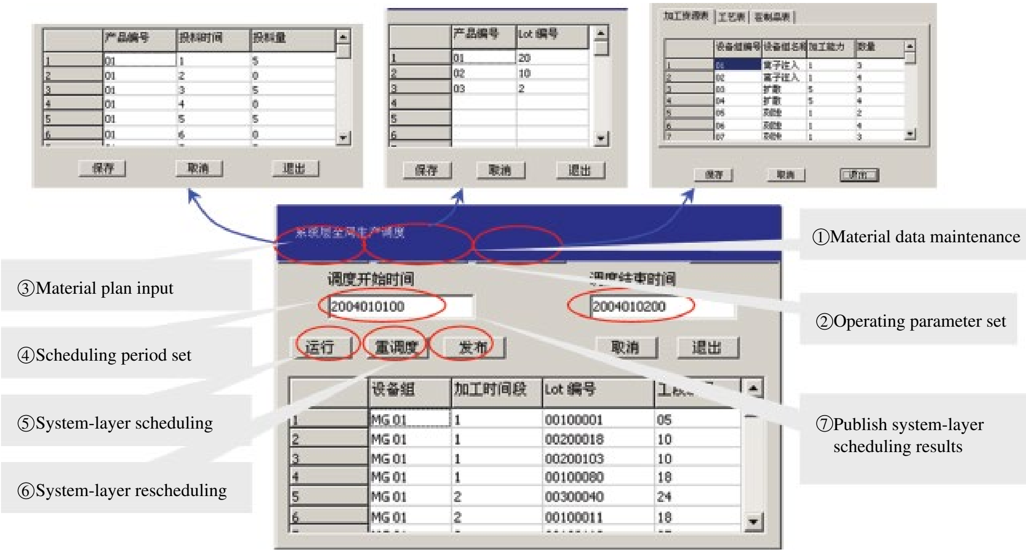

Figure 10-14 The system-layer global collaborative control module based on Combinatorial auction.

7) *System-layer global rescheduling result publish:* send system-layer global rescheduling results to the ETAEMS/GPGP-CN-based production scheduling module at the machine group layer.

At the machine group layer, ETAEMS/GPGP-CN-based production scheduling extends non-local effects in TAEMS to quantitatively describe and analyze collaborative relationships in the production scheduling process. Meanwhile, the collaborative production scheduling process at the machine group layer is achieved by collecting upstream and downstream collaborative information with the GPGP mechanism and employing the bidding mechanism in the improved contract net (shown in Figure 10-15). In this procedure, collaborative data of upstream and downstream information is published by the collaborative scheduling Agent.

With respect to the uncertain and multi-valued relationships among various environmental factors in the rescheduling process, the rescheduling optimization module integrates neural networks with the fuzzy sets theory to solve uncertain problems. By learning from training samples, the module identifies and analyzes the uncertain relationships between operational parameters and rescheduling policies. When the environment or the manufacturing system changes, the rescheduling optimization module selects optimized rescheduling policies in accordance with operational states of manufacturing systems, and then sends them to the Multi-Agent production scheduling system (shown in Figure 10-16) so as to ensure the stability of manufacturing systems.

10.5.6 Production Control in the Manufacturing Process

The Multi-Agent production control system mainly completes the production dispatching process. And it also tracks execution of manufacturing processes to provide real-time accurate data for production control and further ensure the traceability of manufacturing processes. Through manufacturing process management, the process quality data, equipment data, production data and other data are analyzed and visualized (Figure 10-17).

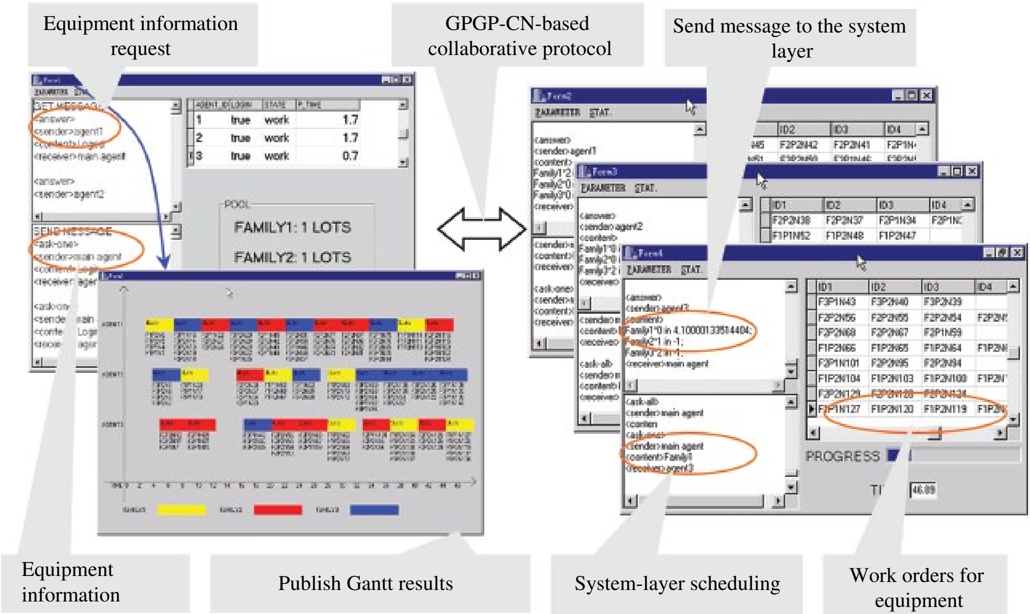

Figure 10-15 Results of the ETAEMS/GPGP-CN-based collaborative dynamic control module at the machine layer.

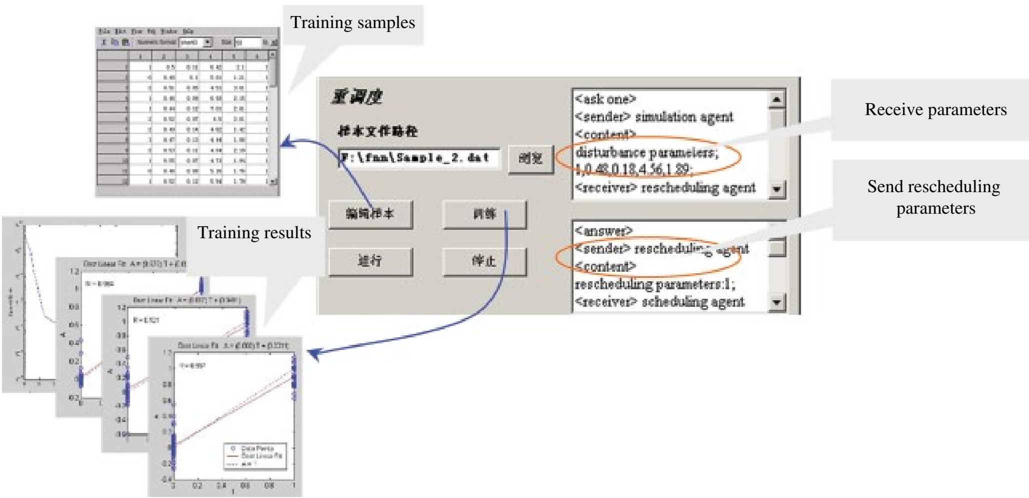

Figure 10-16 Rescheduling results of the Multi-Agent production scheduling system.

Figure 10-17 Multi-Agent production control system.

1) *Production dispatching:* the Multi-Agent production control system accepts production plans published by the Multi-Agent production scheduling system and generates the work order to instruct the operations of machines.
2) *Data acquisition:* the Multi-Agent equipment data acquisition system and the Multi-Agent material data acquisition system collect equipment data and material data, these data are visualized in order to generate the records of equipment and material commands.
3) *Production process management:* The material and equipment operating processes are visualized to display alarm messages when equipment or quality abnormality occur. Meanwhile, production process records are checked according to material number, which includes equipment status, service numbers, time and location of data collection, equipment names, and so on.

10.6 Conclusion

The hardware and software architecture of an Agent-based production planning and control prototype system has been analyzed in this chapter. A simulation test platform has been developed as well. The effectiveness of the proposed Agent-based production planning and control method has been demonstrated through this application.

References

[1] Zhang Jie, Wang Yu, Liu Shiping, et al. Multi-Agent Frame Shop Floor Control System. Huazhong University of Science and Technology, 2001, 29 (11): 9–12.

[2] Liu, S., Zhang, J., Rao, Y., et al. Construction of Communication and Coordination Mechanism in a Multi-Agent System. *17th International Conference Computer Aided Production Engineering*. Wuhan, 2000, 10, 28–30.

[3] Xiong Guangleng. *Continuous System Simulation and Discrete Event Simulation*. Beijing: Tsinghua University Press, 1999.
[4] Guo YunFang. *Computer Simulation*, Beijing: Beijing University of Aeronautics and Astronautics Press, 1991.
[5] Zhou Yan and Dai Jianwei. *HLA Simulation Program Designed* Beijing: Electronic Industry Press, 2002.

Index