UNIVERSITÉ DU QUÉBEC À CHICOUTIMI

MÉMOIRE PRÉSENTÉ À
L'UNIVERSITÉ DU QUÉBEC À CHICOUTIMI
COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN INFORMATIQUE

OFFERTE À

L'UNIVERSITÉ DU QUÉBEC À CHICOUTIMI

PAR

LI-LONG

AN INTELLIGENT TUTORING SYSTEM BASED ON AGENT

JUNE 2009

ABSTRACT

In this paper, the author reviews the history of Intelligent Tutoring System (ITS) and discusses their development in the context of Artificial Intelligence (AI) and educational theory.

This paper also introduces the relatively knowledge of Intelligent Agent （IA） and Multi-Agent system, such as the background of the appearance of Multi-Agent system, the commutation language and system of agents, the correspondence model of Multi-Agent system and the collaboration model of Multi-Agent system. And introduces a Multi-Agent system on the base of WWW, the structure of the system and the design of each components are also mentioned in this paper.

Based on this Multi-Agent system, the author brings in an example —— an intelligent tutoring system based on agent. In this part, the component and the structure of the system are presentation, especially the core of the system —— tutor agent's function and it's realization.

At the end of this paper, the author concludes the thesis and analyses the use of agent in ITS. And finally talks about the future trend of the ITS.

Keywords: Artificial Intelligence (AI), Computer simulation, Intelligent Tutoring System (ITS), Multi-agent. Intelligent Agent

ACKNOWLEDGEMENT

I wish to express my sincere gratitude to my research supervisor, Prof. Cheng-Ming who helps and guides me on my academic and professional success. He provides me with substantial amount of help and suggestions throughout my work. Special thanks should be given to Prof. Paul for being a member of my thesis committee. I thank him especially for his key instruction at the opening speech stage of my thesis and the labor check at the ending stage.

I would also want to thank my assistants Wang-Bin and Lin Gui-Nan. They give me many helps in the researching. I will cherish the memory of the years staying with them forever.

Thanks to my parents for their unending support and eagerness to love.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ACRONYMS

AI	Artificial Intelligence
ACL	Agent Communication Language
ANN	Artificial Neural Networks
AOP	Agent Oriented Programming
C/S	Client/Server
DAI	Distributed Artificial Intelligence
DMAS	Distributed Multi-Agent System
DPS	Distributed Problem Solving
IA	Intelligent Agent
ITS	Intelligent Tutoring System
KIF	Knowledge Interchange Format
KQML	Knowledge Query Manipulation Language
MAS	Multi-Agent System
MIT	Massachusetts Institute Technology
NLP	Natural Language Processing
OECD	Organization for Economic Co-operation and Development
OOP	Object-Oriented Programming
OOT	Object-Oriented Technology
PNP	Plug-and-Play
VR	Virtual Reality

CHAPTER 1

Introduction

1.1 Background and Purpose of Research

In recent years, higher education enrollment in China has been expanding rapidly. Great individual differences exist among college and university students. Students are influenced by different living environments, educational background and even their subjective initiatives, so they are more likely to be different in their capabilities. If teachers ignore the individual differences and simply adopt the cramming method of teaching, they can hardly achieve the ideal. Take the Computer Fundamentals Teaching Course as an example. Freshmen from different parts of the country usually have experienced different senior high-school education. Students from remote countryside have very rarely used computers, while those in the city are quite familiar with the basic computer operations. Therefore, it is difficult to solve the problem of individual differences if too many students have their class in one classroom. This monotonous teaching method is bound to affect the positivity of some students as well as the teaching effects.

The intelligent tutoring system (ITS) is any computer system that provides direct customized instruction or feedback to students, i.e. without the intervention of human beings, whilst performing a task. [1] Thus, ITS implements the theory of learning by doing. An ITS may employ a range of different technologies. However, usually such systems are more narrowly conceived of as artificial intelligence systems, more specifically expert systems made to simulate aspects of a human tutor. Intelligent Tutor Systems have been around since the late 1970s, but increased in popularity in the 1990s.

In china many teachers and students have a strong desire to use the Intelligent Tutoring System (ITS). It can replace or partly replace teachers and teach students according to their different characters and personalities.

Though the research of ITS in China starts late, it has been developing rather quickly in recent years. Many scientific institutions have studied ITS and tried to further its development. However, intelligentization of ITS is not as developed as that in the west, and the teaching based on these systems can hardly provide individualized tutoring or instruction

This thesis tries to introduce the trend of development of ITS based on the current research. And also introduces the agent technology in the AI field. Based on the advantage of the agent technology, the thesis put forward an idea of ITS using agent technology. That is the main content of the thesis: An Intelligent Tutoring System based on Agent. The independent and cooperative relations between agents feature largely the multi-agent system. And these features will help solve the complicated problems in ITS and support the realization of the complex functions of the teaching system. Finally, Distributed Multi Agent System (MAS) will be accomplished in terms of the features of agents. The application of this system will reinforce the intelligentization of ITS and help achieve the final goal.

1.2 The use of Agent in Intelligent Tutoring System （ITS）

With the social development, Intelligent Computer Assisted Instruction System, also called the Intelligent Tutoring System (ITS), has aroused much attention. Many kinds of ITS have been developed home and abroad. Network engineers from all other countries are conducting wide and further research on ITS and have indeed made great achievements.

Nevertheless, in China, the development of ITS lags far behind and cannot provide effective teaching. This leads to the low effectiveness and less of attraction of the network teaching system. Therefore, the task of top priority now is to improve the individualization and intelligentization of ITS.

In this thesis, the elaboration of agent and multi-agent system, the popular techniques in the field of Artificial Intelligence (AI) has been given priority. There is also a detailed introduction about the features and application of these techniques. In 2.5, a conclusion that the use of agent techniques may improve the intelligentization of ITS has been reached. This thesis tries to make full use of the multi-agent system to put forward a Distributed

Multi Agent System (DMAS). All this is based on features of agents, whose cooperative relations can help solve the complicated problems in ITS. The application of the DMAS in the current network teaching can help greatly improve the ITS and promote its intelligentization. The system thus will be more humanized and individualized.

This thesis mainly adopts the Multi-Agent techniques in Distributed Artificial Intelligence (DAI). A multi-agent system consists of many agents. It uses more than one agent to solve more complicated problems. These agents can cooperate with each other in a harmonious way. To make the cooperation between agents more effective, the relationship between these agents becomes the essential point in the research. Generally speaking, each agent is considered a physical or abstract entity. It can act on itself and its surrounding environment. At the same time, it can communicate with the other agents. The basic approach of DMAS is to improve the capability of computer in solving complicated problems by imitating the operating systems of the human society. Another feasible way to enhance the capability of the whole system is the cooperation of many agents. In this way, the task will be achieved by subdivision or simply through cooperation. In addition, by means of the cooperation of many agents, the system can overcome such shortcomings like the incompleteness of a single agent, and the inadequacy in dealing with the data.

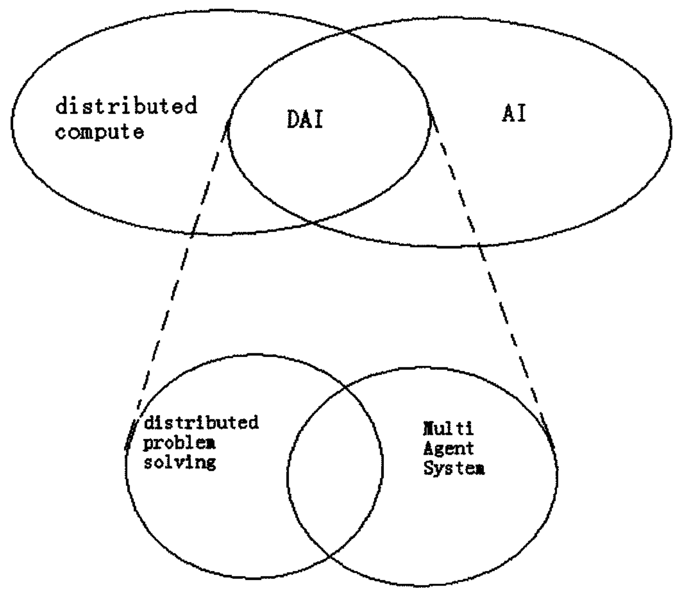

Fig.1 Multi-Agent System & Distributed Artificial Intelligence

The research of multi-agent system is one of the research directions of Distributed Artificial Intelligence (DAI), and their relations are showed clearly in the figure above.

DAI is an important branch of Artificial Intelligence. It consists of many agents and each agent act in a semi-autonomic way. Distributed Problem Solving (DPS) and multi-agent system are the two major aspects in the research of DAI system. Distributed Problem Solving mainly focuses on how a distributed network of semi-autonomous processing nodes work together to solve a single problem. Multi-agent system refers to the fulfillment or realization of some task through the cooperation of many agents in an autonomous way. It focuses on the effective interaction between agents. It aims at achieving a better interaction among the agents through their cooperation during problem solving. At present, study of the development of MAS includes theories of multi-agent systems, communication technology and interactive techniques, architecture and organization, Agent Oriented Programming (AOP), the cooperation and coordination between agents, etc.

This thesis mainly introduces the development of multi-agent system, the basic concepts, the features of agents, the relevant theories and approaches, as well as the cooperation and coordination between agents. Finally, this thesis will introduce the application of multi-agent system in network-assistant teaching.

1.3 Thesis Organization

Chapter 2 introduces briefly the Intelligent Tutoring system (ITS). This chapter describes the development of ITS. The main idea is to improve the system performance with the help of artificial intelligence technology. Finally, this chapter illustrates in detail the use of agents in the distance teaching system.

Chapter 3 covers the development of the agent and multi-agent as well as the basic concepts. 3.1 examine the background of the appearance of agents. The reasons why software agents appear can be summarized as follows: 1.The simplification of the complexity of Distributed Artificial Intelligence (DAI); 2.Overcoming the deficiencies in human-computer interface. 3.2 introduce in detail the concepts, features and architecture of an agent. 3.3 deal with the concepts, features of the multi-agent system (MAS) and its application as well. This chapter makes it easy for readers to acquaint themselves with the agent technology in artificial intelligence.

Because this thesis mainly focuses on the research and application of the MAS, Chapter 4 further deals with the theories and technologies of MAS. MAS needs to borrow from some relevant theories and technologies, which can mainly be summarized as follows: knowledge representation and reasoning, agent communication language, communication and interaction technologies, agent-based collaboration models, agent-based negotiation models, agent architecture and organization, and so on.

Chapter 5 puts forward the architecture based on the models mentioned in Chapter 4. Furthermore, it studies the user interface, user agent, application agent and control agent.

Chapter 6 uses the agent technology in the ITS. This is based on the features of the agent and ITS. This chapter also adopts simulation modeling to analyze the components of the system and the function and architecture of each agent in the model.

Chapter 7 the author concludes the thesis and analyses the use of agent in ITS. And finally talks about the future of the ITS.

CHAPTER 2

Intelligent Tutoring System

In recent years, Intelligent Tutoring system (ITS) has attracted much attention in Technology of Education in China. This chapter illustrates the basic concepts and the development of ITS and then predicts the future trend of ITS. By analyzing the application of agents in Distance Learning Assistant, this chapter makes it clear that ITS will have a bright future and requires further studies.

2.1 What is ITS?

Artificial intelligence in education comes of age in systems now called "intelligent tutors", a step beyond traditional computer-assisted instruction. Computer-assisted instruction evolves toward intelligent tutoring systems (ITS) by passing three tests of intelligence. First, the subject matter, or domain, must be "known" to the computer system well enough for this embedded expert to draw inferences or solve problems in the domain. Second, the system must be able to deduce a learner's approximation of that knowledge. Third, the tutorial strategy or pedagogy must be intelligent in that the "instructor in the box" can implement strategies to reduce the difference between expert and student performance.[2]

ITS covers Artificial Intelligence, computer science, education, psychology, behavioral science, and many other fields. The study of ITS is to use computers more widely in education and realize the intelligentization of a computer system. In this way, the computer can replace teachers in some degree and help to optimize teaching and learning.

ITS is a comprehensive subject and is closely related to AI, computer science, cognitive science, noetic science, education, psychology and behavioral science. Its final aim is to make the computer shoulder the responsibility of human education, namely, to realize the intelligentization of a computer system and optimize teaching and learning by replacing teachers. The significance of the study of ITS is to reduce teachers' amount of work and reach high standards of teaching. This requires that the computer system consist of domain knowledge, teaching knowledge, the capabilities of learner-computer interaction. The major features of this system are as follows:

(1) It can automatically provide all kinds of exercises and rehearsal practices.

(2) It can adjust the learning content and rate of progress according to the abilities of different learners.

(3) It can solve problems on the basis of its comprehension of the teaching materials.

(4) It can understand and respond to natural language, thus form a free question-answering system and improve the interaction between learners and computer.

(5) It possesses the ability to explain the teaching materials.

(6) It can judge learners' mistakes, analyze the reasons and correct it in time.

(7) It can appraise learners' learning behavior.

It is difficult for the ITS to possess all the features mentioned above though it will be perfect to do so. Therefore, ITS need merely possess one or several of the features required.

2.2 The history of ITS

ITS appeared in the 1970's when Bolt Neranek Newman Inc. developed the Scholar system. The Scholar system has been regarded as the earliest ITS. It adds different functions into ITS, such as the judgment of learners' learning behavior and abilities, the training strategy, and so on. At the same time, AI has been used in the function of consultation. This means the system can restore the relevant questions and skills, adopt proper training strategy and provide suitable teaching materials. Thus, a special software system comes into being. It can teach according to learners' different abilities, weakness and their favorite learning styles.

Later on, other systems such as Why、Sophie、West、 Buggy、Neomycin appeared. These systems adopted different AI techniques and applied them to ITS. These techniques include knowledge representation, expert system, problem solving, reasoning approach, and so on. The efforts have been proved fruitful.

The study of ITS in China is rather late. At the very beginning, only several universities and research institutes carried out the research of ITS intermittently. These researches are used only for the sake of demonstration. Few systems undergo strict evaluation and testing. However, ITS has been developing rapidly in recent years. Even some companies have participated in the research program. With its rapid development in China, ITS will definitely play an active role in China's education reform.

2.3 Use in Practice

All this is a substantial amount of work, even if authoring tools have become available to ease the task [3]. This means that building an ITS is an option only in situations in which they, in spite of their relatively high development costs, still reduce the overall costs through reducing the need for human instructors or sufficiently boosting overall productivity. Such situations occur when large groups need to be tutored simultaneously or many replicated tutoring efforts are needed. Cases in point are technical training situations such as training of military recruits and high school mathematics. One specific type of intelligent tutoring system, Cognitive Tutors, has been incorporated into mathematics curricula in a substantial number of United States high schools, producing improved student learning outcomes on final exams and standardized tests [4]. Intelligent tutoring systems have been constructed to help students learn geography, circuits, medical diagnosis, computer programming, mathematics, physics, genetics, chemistry, etc.

In the following two chapters, a further discussion will be made about agent technology and multi-agent system.

2.4 ITS Conference

The Intelligent Tutoring Systems conference was typically held every other year in Montréal (Canada) by Claude Frasson and Gilles Gauthier in 1988, 1992, 1996 and 2000; in San Antonio (US) by Carol Redfield and Valerie Shute in 1998; in Biarritz (France) and San Sebastian (Spain) by Guy Gouardères and Stefano Cerri in 2002; in Maceio (Brazil) by Rosa Maria Vicari and Fábio Paraguaçu in 2004; in Jhongli (Taiwan) by Tak-Wai Chan in 2006. The conference was recently back in Montreal in 2008 (for its 20th anniversary) by Roger Nkambou and Susanne Lajoie. ITS'2010 will be held in Pittsburgh (US) by Jack Mostow and Vincent Aleven. The International Artificial Intelligence in Education (AIED) Societypublishes. The International Journal of Artificial Intelligence in Education (IJAIED) and produces the International Conference on Artificial Intelligence in Education every odd numbered year. The American Association of Artificial Intelligence (AAAI) (www.aaai.org) sometimes has symposia and papers related to intelligent tutoring systems. A number of books have been written on ITS including three published by Lawrence Erlbaum Associates. [5]

2.5 IA in Distance teaching system

In artificial intelligence, an intelligent agent (IA) is an autonomous entity which observes and acts upon an environment (i.e. it is an agent) and directs its activity towards achieving goals (i.e. it is rational).[1] Intelligent agents may also learn or use knowledge to achieve their goals.[6]

1. Individualization of Distance teaching

At present, individualization in distance learning assistant only focuses on time and space. The teaching methods and materials have not been changed virtually. Individual education can be regarded as one of virtues of the Web. However, if just change the classroom teaching into the form of network teaching, individual education can not be really carried out and the virtues of the traditional teaching method are lost. When agent technology is adopted, Intelligent Agent (IA) can choose different teaching methods and teaching resources for each learner. This selection is based on learners' various learning levels, teaching materials, difficulties in learning and learning motivations. Each learner will experience a different learning process. The agents used in the system will become "private tutors" for each learner.

2. Human-computer interaction and teaching method

With the introduction of agent technology, the human-computer interaction will undergo a radical transformation. The essential feature of the network teaching based on multi-agent is its great "kindness" towards learners. During the teaching process, learners will study in a completely different environment, i.e., the computer can "hear" the learners' voice and accordingly adjust the whole teaching process.

3. Self-adaptive evolution of agent

When self-adaptive evolution of agent is introduced into education, the potentiality of the system is optimized. The agents provide different learning patterns to each learner. But they will continue to change in accordance with the human-computer interaction. Deep acquaintance with the learner will make the system more adaptable to learners' requirements. In some degree, the quantity and quality of the teaching resources mean the adaptability of the system to learners.

4. Efficient use of teaching resources

At present, teaching resources available on the Internet are in abundance. However, they are almost in a mess and can only be regarded as some kind of information or data. What learners want is knowledge. It is a great, complex task to transform all these information and data into useful knowledge topics. In the current situation, many learners are "drowned" when they try to utilize the resources on the Internet. They just cannot use the resources efficiently. In ITS based on multi-agent, the agents will help transform what is need by learners. Accordingly, all the information and data available on the Internet will be fully used by learners.

5. Cooperation and intelligentization

In ITS based on multi-agent, questions in the discuss board will be summed up and special topics will be further discussed. Learners can understand those questions more thoroughly. This shows the virtues of cooperation in ITS.

6. Networking teaching helps cultivate learners' abilities and develop quality education

Learners may bring forward their own questions. With the help of ITS, they can solve their problems step by step according to what have been discussed. They can even further their research. Instead of a passive teaching model, ITS based on multi-agent provides exhaustive resources and an open environment. Learners themselves can decide which part is more important in their study and which learning pattern they prefer. In the learning process, all questions will be solved gradually and learners' abilities in solving problems will be improved at the same time.

All in all, ITS base on multi-agent has a promising prospect in network teaching.

This thesis mainly focuses on the application of agent technology in ITS. More details are provided in the following part.

Agent technology is a hot topic in AI at present. People have been trying to use the agent technology to unify and further develop AI. Some even try to use it to unify and develop software. This shows that agent technology has a wide and promising application prospect. Thus in the application of ITS, we can utilize the multi-agent system. This is mainly because the independent and cooperative relations between agents feature largely the multi-agent system. And these features will help solve the complicated problems in ITS and support the realization of the complex functions of the teaching system.

In the following two chapters, a further discussion will be made about agent technology and multi-agent system.

CHAPTER 3

Agent and Multi-Agent System

3.1 Background of Agent

The original meaning of the term *agent* is "representation", namely, someone who represents somebody else or an organization in doing something, and sometimes it refers to the means by which an effect or result is produced. The term *agent* has appeared long before in business as a go-between. It is used to denote an initiator of action and independent decision-making. People believe agents can achieve the task required by adopting actively some kind of approaches. Though agents were used especially in business before, now they are widely related to computer science. After years of study, researchers give the term *agent* different definitions. So it has obtained different connotations accordingly. The meaning of *agent* nowadays is quite different in many important aspects from the past. However, in comparison with its functions as a go-between in business, its essence and basic functions have changed little.

People have been longing for non-human agents for such a long time. But not until World War II, with the development of computer technology and Control Theory, did similar autonomic agents appear. Its ancestors, to which agents today are closely related, are probably some basic control devices, such as the Product Line Director in a workshop, automatic take-off and landing control, and so on. If robots can be called "hardware agents", then software agents will form a "software robot", which will fulfill its mission in the computer world.

The research and application of agent technology in computer science originates from Distributed Artificial Intelligence (DAI). In the 1970s, MIT (Massachusetts Institute of Technology) researchers made a thorough study of DAI. When they were analyzing how the information system solved complex problems, they found that collaboration between some simple information and even some systems would obviously improve the capacities of the whole system. In addition, a feasible cooperative mechanism could improve the intelligence of the system as a whole. As a result, concepts and approaches in the study of agents are developed. These agents possess certain intelligence. They can passively respond to the requirements of information disposal. Except completing the task required, the agents can predict and adapt to users' interests, and they can even actively seek methods to meet their needs.

At the very beginning, software agent researchers mainly studied the computation model of DAI, and proposed two questions from a practical point of view. One is to simplify the complexity of DAI, and the other is to overcome deficiencies in human-computer interface. Solving these two problems aims at making a further study of human-computer interface. On the one hand, it tends to develop a more effective, universal and consistent computer interface between agents, so as to replace the current computer interface between programs. In this way, it can help realize the abstraction and encapsulation of implementation details of computer hardware, software as well as communications model. On the other hand, it hopes to use agents to meet users' needs, thereby simplifying the human-computer interface (user interface) designed to perform complicated tasks.

All in all, the reasons why software agents appear can be summarized as follows:

1. The simplification of the complexity of DAI

Human society can be regarded as an environment in which there are many individuals, who have their own aims, wishes and instincts. In this environment, the old dies out and new ones are continually generated. During their whole lives, individuals compete and sometime cooperate with each other to achieve their own goals. This promotes the development of the whole environment from the macro point of view. With the diversity of requirements and the complexity of computation, most people believe in the future computation environment will evolve towards simulation of human behavior. It will be a distributed and open computation environment, consisting of many distributed software systems running on a variety of platforms. At present, most of the software systems are separated from each other, though sometimes they do communicate and collaborate with each other on some basic aspects, such as file transfer, printing service, database query, etc.. Furthermore, many software systems can only be connected each other through special interface. In recent years, with the development of Object-Oriented Technology (OOT) and other Web standards like TCP/IP , HTTP , ODBC, encapsulation begin to appear in the connection between systems or services. Figure 2 describes the changing process of the connection between systems. [7]

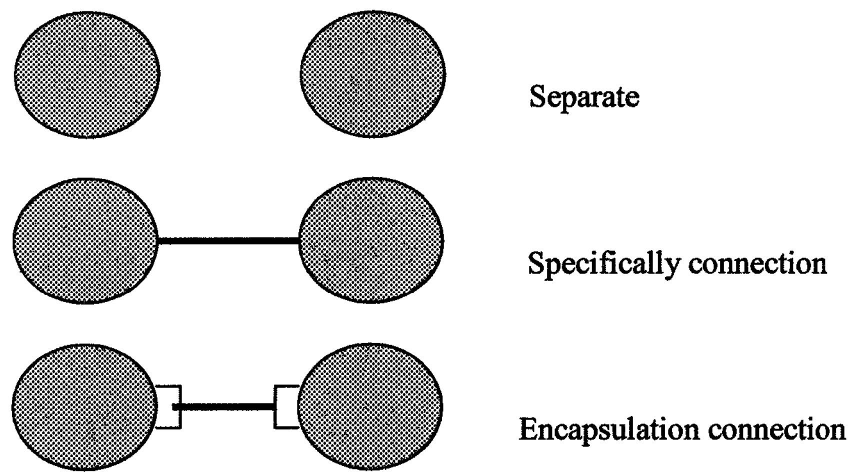

Fig.2 The changing process of the connection between systems

With the transference from distributed computing based on network operation system (NOS) to distributed computing based on INTERNET and INTRANET, interoperation of Web services starts to appear. It is independent and supports distributed computing based on INTERNET and INTRANET, such as the Naming Service, Directory Service, Security Service, and so on. Some software and hardware manufacturers are trying to formulate a set of network service protocols independent of operating system. They also intend to develop the relevant application interface. When the protocols and interface are widely used, any operating system installed on the computer will be available to the whole Internet, that is, integration and interoperation will occur between the existing system and other systems. This is also called "Plug-and-play (PnP)".

Interoperation of high-tech software requires that the software system possesses the capabilities and knowledge to define and describe the system itself. Only in this way can the software system automatically perform many functions during collaboration, including achieving the tasks required, distributing resources, executing instructions, monitoring and possible intervention. From this point of view, AI plays an important role in the management of all resources. Nevertheless, though a single agent may function in a miniature network system, it becomes valid when more systems are involved. Thus the capacities of each single may become the bottleneck of each system. Intelligent interoperation refers to the application of many agents to fulfill the task required in each collaborative system. Moreover, intelligent interoperation asks agents to reflect the user's intention at a higher level instead of the practical realization. Therefore, it should not only possess the capsulation ability in basic communication, but have more intelligent abilities in the process of capsulation.

2. Overcome the deficiencies in human-computer interface

At the present time, human-computer interface can be operated directly by the user. This has become one of the basic functions of the computer software. It is a big progress for the human-computer interface which is famous for its commanding lines. Through the use of direct human-computer interface, the user can operate the software by way of visualization. However, the user will be continually informed which object he or she may deal with. When complex or large-scale tasks are involved, the deficiencies in human-computer interface are exposed. It brings great inconvenience to the user. Approaches based on agents can mend such deficiencies. This is indicated in the following part:

- Information search: Direct human-computer interface can search a massive amount of information. In large-scale distributed systems, it is hard to find out information needed simply by browsing or traditional index approaches. Agents can be operated behind the scenes. Its search and filter functions will help the user obtain large amount of information or resources.
- Operation time: Direct human-computer interface can only respond to the user's real-time demands. But sometimes the user probably needs to plan his or her demands first, then carries it out at a special time in the future. Or the user may hope the software can respond automatically to other requirements when he or she

is absent. Here agents can again be used to help perform the task required at a particular time, or make the system respond automatically.

- Expanding function: Most of the direct human-computer interfaces can not obtain more functions through the combination of action and object. However, agents can help form an expansible or reconfigurable interface. It can not only execute instructions, but also communicate with the user. Thus the task will be finished accordingly.
- Convenient operation: It is quite easy to use direct human-computer interface to achieve simple tasks. But for complex ones, it appears to be too rigid. The introduction of agents may create more favorable conditions for solving complex problems.
- Range: Direct human-computer interface mainly focuses on the functions of software instead of the user tasks and environments. With the help of agents, the user tasks and other conditions can all be taken into consideration.
- Learning ability: Direct human-computer interface can not improve itself according to user's learning behavior. Agents can help better the computation and operation through learning behavior.

To meet the needs mentioned above, agents are further developed and applied in more and more fields. An agent can be regarded as an intelligent entity. It can control its behavior by predefined programs. At the same time, it can adjust its own behavior according to its interaction with the surrounding environment. This shows its autonomy, reactivity, initiative and agility, which will surely lead to the fulfillment of the task. Since its internal structure is complicated, the speed of an agent is comparatively slow. With the accumulation of knowledge and its fast speed of updating, a single agent is obviously not enough. In recent years, researchers have proposed the conception of multi-agent, which enables agents get the learning capacity. Through collaboration as well as communication between agents, multi-agent system can provide superior speed and simplicity, and can achieve the tasks with ease.

3.2 Basic concepts concerning agents

3.2.1 Definition of agents

Up to now, the term "agent" has not had a unified and authoritative definition. Researchers have given "agent" definitions of their own based on their field of studies.

Intelligent agents are often described schematically as an abstract functional system similar to a computer program. For this reason, intelligent agents are sometimes called abstract intelligent agents (AIA) to distinguish them from their real world implementations as computer systems, biological systems, or organizations. Some definitions of intelligent agents emphasize their autonomy, and so prefer the term autonomous intelligent agents. Still others (notably Russell & Norvig (2003)) considered goal-directed behavior as the essence of intelligent and so prefer a term borrowed from economics, "rational agent".[8]

The following definition is by Foundation for Intelligent physical Agent (FIPA):

An agent is an entity in certain environment. It can perceive its environment and react to it. An agent can be regarded merely as software or hardware supported by some particular software systems.

Distributed Artificial Intelligence (DAI):

According to DAI, An agent is an object-oriented software entity. It functions continuously and autonomously in a particular environment, often inhabited by other agents and processes. “It personalizes the description of an agent. That is, an agent is an entity involving belief, promise, obligation, intention and other mentalities”.

Software Engineering:

Software Engineering tries to provide an agent-oriented modeling approach for distributed network. “An agent-oriented modeling approach adopts more abstract concepts in order to describe actually a more complex concurrent system. Like object-oriented approach, it is a means to observe the world and solve problems. "

While in distributed computation and computer network, an agent is mainly regarded as a customer facing entity or server entity performing special tasks.

Here, we have mainly discussed the agent. According to the different definitions given above, we try to give a definition of the agent which will be universally accepted in some degree: an agent is a software entity which can function continuously and autonomously in a particular environment. It can perform a certain task independently. And at the same time, it can collaborate with other agents to achieve the goal.

3.2.2 Features of an agent

An agent is a software entity based on internal drive. It is autonomic, adaptive, interoperative and intelligent. It can react to itself as well as its environment. It can make an adaptive response to changes. Here are the main features of an agent: [9]

(1) Interaction

Interaction is the primary feature an agent should possess. It includes two major aspects: its interaction with the user and other agents. An agent should have the ability to interact with the user. It can receive instructions from the user and then help fulfill the tasks required. Meanwhile, it can interact with other agents. It can improve its pertinence, agility and the efficiency by further analysis. The implied meaning of interaction is that an agent should be active.

(2) Autonomy

This is the personified feature of an agent. It aims to adapt to the massive and complex distributed network environment. Autonomy is one of the basic capabilities an agent should possess. An agent should be able to perceive its environment and react timely to predicted or non-predicted events. This reaction belongs to its own computation resources or control mechanism. And it is decided by the internal mechanism of an agent.

(3) Adaptivity

An agent can make a plan according to its target and environment. It can adjust its plan according to the changing environment.

(4) Purposefulness

An agent is intensely purposeful. It can adopt certain methods according to its principles to achieve the goal.

(5) Collaboration

The target an agent can realize is usually a part of the target set. As a result, it has to consult and collaborate with other agents through the process of sharing information.

(6) Intelligence

An agent can perceive its internal state as well as its environment. It can carry on the process of perception—reasoning—action through its perceptron and actuator. Intelligence can be achieved by way of AI programming.

Intelligent agents have been defined many different ways. According to Nikola Kasabov IA systems should exhibit the following characteristics : [10]

- Accommodate new problem solving rules incrementally.
- Adapt online and in real time.
- Be able to analyze itself in terms of behavior, error and success.
- Learn and improve through interaction with the environment (embodiment).
- Learn quickly from large amounts of data.
- Have memory-based exemplar storage and retrieval capacities.
- Have parameters to represent short and long term memory, age, forgetting, etc.

AI researchers generally believe that an agent should not only possess the features mentioned above, but should also acquire some concepts which merely belong to the human society. These concepts include knowledge, belief, aim, obligation, etc. Shoham claims that an agent is an entity consisting of many mentalities (such as belief, ability, choice and promise). Some other AI researchers further consider the emotion-based agent. An agent also possesses other features, such as mobility, veracity, benevolence and rationality.

In fact, an agent can show more features, though there has not been a normal standard till now. This is because an agent is multidisciplinary, and it involves AI, distributed computation, Web, database and many other fields. Therefore, the features of an agent can be discussed from different angles.

3.2.3 The architecture of an agent

The architecture of each agent denotes its basic ingredients, functions, relations between different ingredients, human-computer interaction, computation, etc. Currently, the architecture of a single agent can be grouped into three broad categories:[7]

(1) Reasoning architecture. It includes the explicit symbolic model. It makes its decision through logic reasoning, pattern match and symbol operation. Two major problems exist in this architecture: firstly, it is quite difficult to change the real world into an accurate and appropriate symbol; secondly, it is hard to carry out the processes of reasoning and decision-making of relevant information.

(2) Reactive architecture. It does not include the world model represented by some symbols. It does not adopt the complex symbolic reasoning. On the contrary, it simply reacts to the outside world when its internal state and external environment are appropriate. However, when the agent needs the knowledge which can only be obtained by reasoning or memory, this architecture again faces its own unsolvable problems. Nevertheless, it plays a leading role in the

distributed system at the present time.

(3) Composite architecture. A composite architecture comes into being when the two architectures mentioned above are combined together to avoid their drawbacks.

No matter which architecture is adopted, we find that the architecture of an agent mainly consists of several basic parts (see Fig. 3):

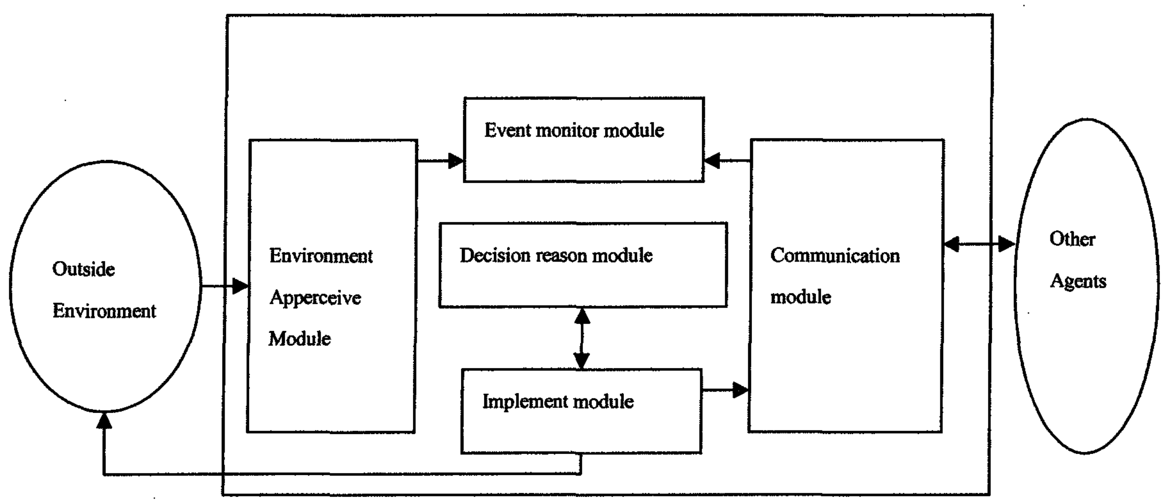

Fig.3 the basic architecture of a single agent

3.3 The Multi-Agent System

3.3.1 The concepts of Multi-Agent System

The multi-agent system is a system consisting of many agents. It aims to solve complex problems which are unsolvable when only one agent is involved. It is a problem solving network which functions through the collaboration between many agents. To make the collaboration between agents more effective, the relationship between these agents becomes the essential point in the research. Generally speaking, each agent is considered a physical or abstract entity. It can act on itself and its surrounding environment. At the same time, it can communicate with other agents. The basic approach of DMAS is to improve the capability of computer in solving complicated problems by imitating the operating systems of the human society. For example, usually one single person can not complete a complicated and massive task. Likewise, it is difficult for a single agent to solve many problems. Therefore, a feasible way to enhance the capability of the whole system is through the collaboration of many agents. In this way, the task will be achieved by subdividing it into smaller parts or simply through collaboration between agents. In addition, by means of the collaboration of many agents, the system can overcome such drawbacks like the lack of information of a single agent, and the inadequacy in dealing with the data with one agent, etc..

The agents in a multi-agent system can collaborate with each other either tightly or loosely. The agent granularity of a multi-agent system can be either coarse or fine. Take Artificial Neural Networks (ANN) as an example. ANN has been regarded by some researchers as a multi-agent system in which agents collaborate tightly with each other. And distributed problem solving system usually consists of a set of agents loosely connected. Each agent can play a part in complex problem solving. It can change its behavior according to its environment, and communicate and collaborate with other agents.

3.3.2 The architecture of Multi-Agent System

Generally speaking, the architecture of a multi-agent system refers to the communicative model and control model between agents. It can influence the capabilities of the whole system. From this point of view, the architecture of the multi-agent system can be grouped into three categories: centralized architecture, distributed architecture, and mixed architecture: [7]

1. Centralized architecture (see Fig. 4): It subdivides the whole system into many groups. Each group adopts centralized control, namely, each group uses a single control agent to coordinate the actions of the remaining agents to accomplish a particular goal, such as task planning and task distribution. Besides, a single agent

will be chosen to take charge of the transfer of information. And the whole system will manage those agent groups in the same way. A centralized architecture can keep the information consistent in the system. Thus it is easier to manage and control the system. This approach also has its own drawbacks. Since agents are becoming more complex and dynamic, the associated control problem becomes much more difficult to be solved efficiently. This process also suffers from another fatal shortcoming: since the control agent manages part of or the whole system, once it fails, the whole system or the part it controls will fail as well.

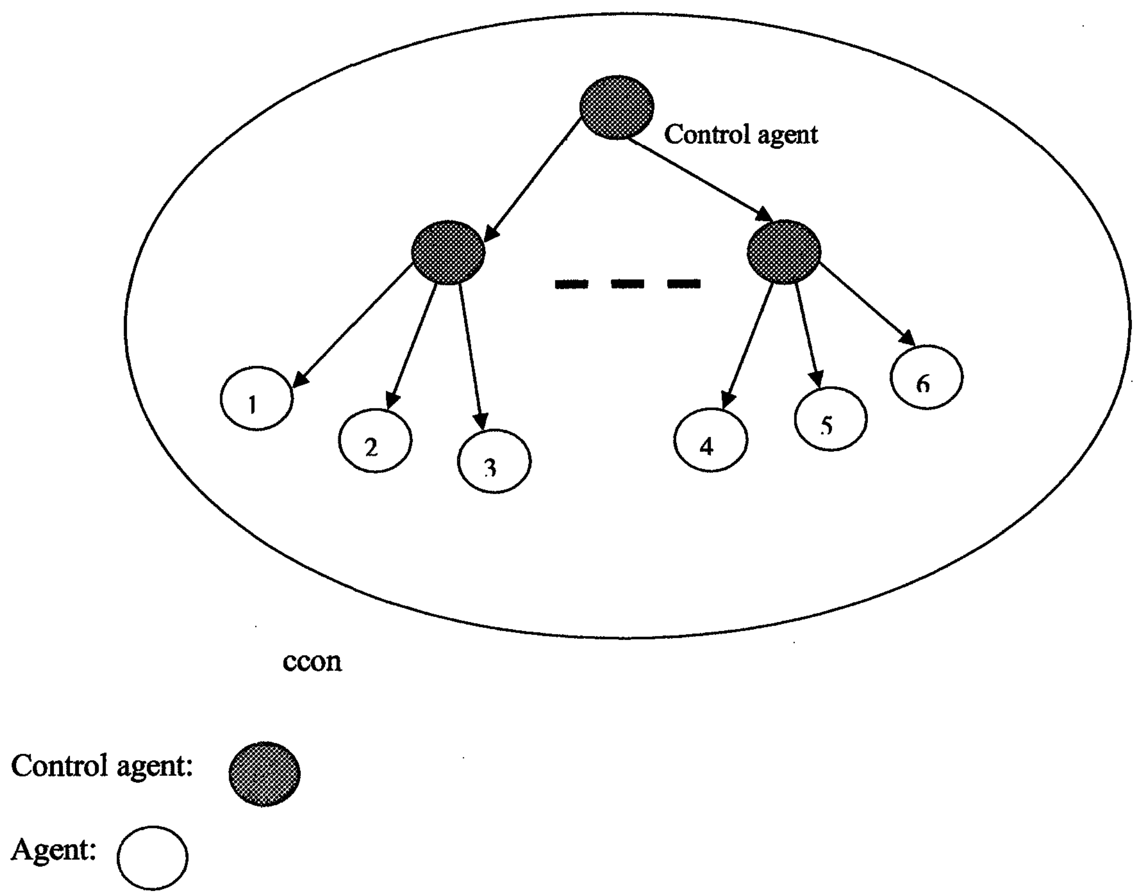

Fig.4 The centralized architecture of the multi-agent system

2. Distributed architecture (see Fig. 5): The distributed architecture lies in between the agent groups and those agents within one group. These agent groups or agents are equal, and no one is more important than the other. Whether an agent can be activated and what behaviors it may adopt depend on the system itself, the environment, the agent state and data currently possessed. Many a medium can exist in this architecture, and they provide necessary services to other agents when they need collaboration. This architecture has many virtues. It is agile and stable, and the control problem becomes more easily to be solved. But it still has its drawbacks. Because each agent group or agent is confined to incomplete information (such as part of the goal or plan), it is quite difficult to keep the whole system consistent.

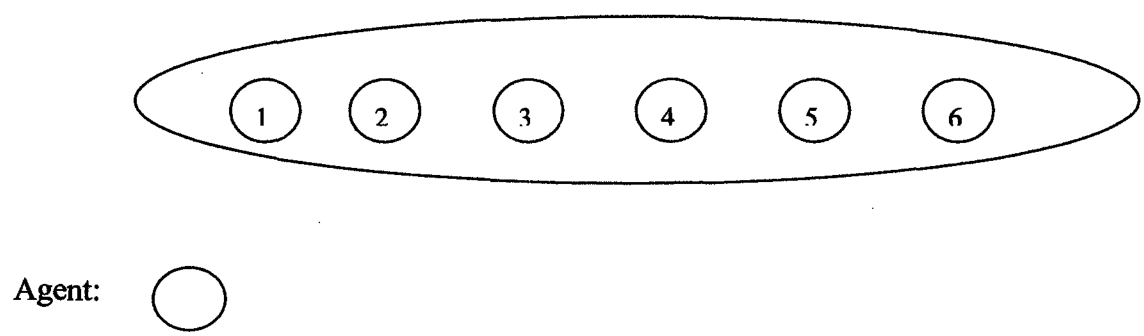

Fig.5 The distributed architecture of the multi-agent system

3. Mixed architecture (see Fig. 6): Generally speaking, it consists of the centralized architecture as well as distributed architecture. It has one or more than one control

agent, which manages certain agents in a special way. It can help solve many problems, such as task distribution between agents, management of sources, coordination, conflict resolution, and so on. All the other agents are equal, and they decide their own behaviors by themselves. This architecture has balanced the advantages and drawbacks of the centralized architecture as well as distributed architecture. It is suitable for the open and complex MAS. Therefore it is most commonly used in MAS.

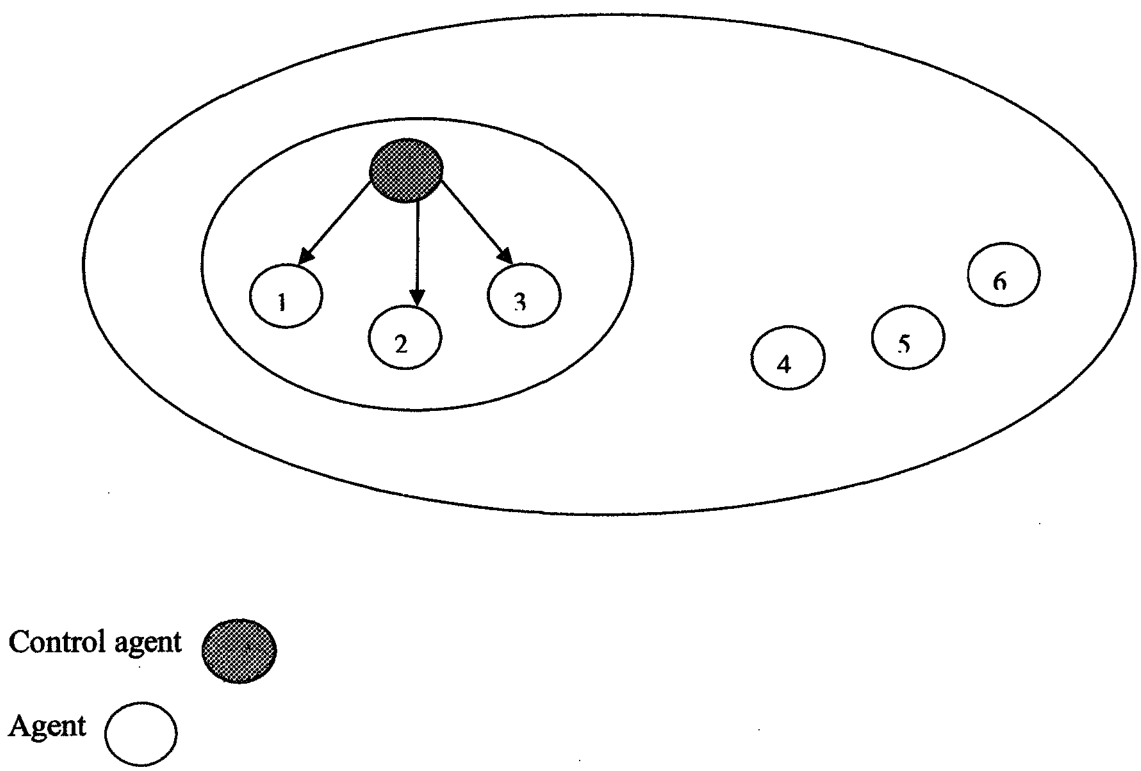

Fig.6 The mixed architecture of the multi-agent system

3.3.3 The application of MAS

At present, MAS theory and technology are widely used in many fields. And it is typically used in Open System and Complex System.

- Open System

  An Open system is flexible. It is a collection of interacting software, hardware, and human components. Its system components are not predetermined and can change according to the events. They can also become heterogeneous components. This is simply because different components are created by different people. And they are developed at different periods based on different technologies. The most typical open system is the Internet, which can be viewed as a huge distributed database. Each node in the network is created by different individuals or groups. Any system concerning Internet has to be able to use these different resources.

- Complex system

  Modularized model can be adopted in the complex system to make it much easier. An agent is quite useful in the modularized model. If the stated task is especially complex, massive and hard to predict, an agent can help subdivide the task into smaller parts. Each simple component can be easily developed and maintained. Moreover, it aims at solving a special problem. In this way, each agent can use the most appropriate method to solve a particular problem. They do not have to use a unified method which may be inefficient.

3.3.4 Types of MAS

Software agent technology has been widely used in mainstream computer science, which generates many kinds of agent-based systems. These systems can be broadly categorized as follows:

Cooperative agent-based system: It is a typical, massive agent-based system. The agent granularity is either coarse. It lays emphasis on the autonomy and interoperation of agents. It performs the task in an open multi-agent environment. And all the agents can be consistent with each other.

Agent-based interface system: This system supports and provides voluntary assistance which is especially helpful to a complex system adopted by the user. It emphasizes the autonomy and learning ability of agents so as to achieve the task required. Its most important feature is that it can be regarded as a personal assistant, who cooperates with the user in the same working environment.

Mobile agent-based system: Agents in this system can roam within the Internet like WWW. They perform the task for the sake of the user. When they fulfill their responsibility, they return to where they belong.

Agent-based information system: Autonomy, adaptability and interoperation feature this system. It can manage, control and collect information in distributed resources.

Reactive agent-based system: In this system, there is no symbolic model of the environment. On the contrary, it reacts to the environment in an impulse-reaction way to show the state of the environment.

In addition, there is also a mixed agent-based system. This system is the combination of two or more than two agent-based systems. A heterogeneous agent-based system is one based on the integration of some different existing agent-based systems.

CHAPTER 4

Multi-Agent System

The multi- agent system need to borrow from some relevant theories and technologies, which mainly include the following parts: knowledge representation and reasoning, agent communication language, communication and interaction technologies, agent-based collaboration models, agent-based negotiation models, agent architecture and organization, and so on.

4.1 Knowledge presentation and reasoning

Knowledge representation is an area in artificial intelligence that is concerned with how to formally "think", that is, how to use a symbol system to represent "a domain of discourse" - that which can be talked about, along with functions that may or may not be within the domain of discourse that allow inference (formalized reasoning) about the objects within the domain of discourse to occur. Generally speaking, some kind of logic is used both to supply a formal semantics of how reasoning functions apply to symbols in the domain of discourse, as well as to supply (depending on the particulars of the logic), operators such as quantifiers, modal operators, etc. that, along with an interpretation theory, give meaning to the sentences in the logic.[11]

There are representation techniques such as frames, rules and semantic networks which have originated from theories of human information processing. Since knowledge is used to achieve intelligent behavior, the fundamental goal of knowledge representation is to represent knowledge in a manner as to facilitate inferencing (i.e. drawing conclusions) from knowledge.

Some issues that arise in knowledge representation from an AI perspective are:

- How do people represent knowledge?
- What is the nature of knowledge and how do we represent it?
- Should a representation scheme deal with a particular domain or should it be general purpose?
- How expressive is a representation scheme or formal language?
- Should the scheme be declarative or procedural?

There has been very little top-down discussion of the knowledge representation (KR) issues and research in this area is a well aged quiltwork. There are well known problems such as "spreading activation" (this is a problem in navigating a network of nodes), "subsumption" (this is concerned with selective inheritance; e.g. an ATV can be thought of as a specialization of a car but it inherits only particular characteristics) and "classification." For example a tomato could be classified both as a fruit and a vegetable.

In the field of artificial intelligence, problem solving can be simplified by an appropriate choice of knowledge representation. Representing knowledge in some ways makes certain problems easier to solve. For example, it is easier to divide numbers represented in Hindu-Arabic numerals than numbers represented as Roman numerals. [11]

Knowledge representation and reasoning is to set up relations between problems existed in the real world and the symbolic reasoning system. The symbolic reasoning system works with the help of domain model. It consists of data structures for information storage and data manipulation. Each relevant element, which is supposed to solve a problem in the multi-agent system, should have its counterpart in the domain model. Such a relevant element can refer to an object or the relations between objects. The mapping relation between the actual model and the domain model enables the agent to reason in the domain model and show the result in the problem domain. Figure 7 shows this abstract mapping process.

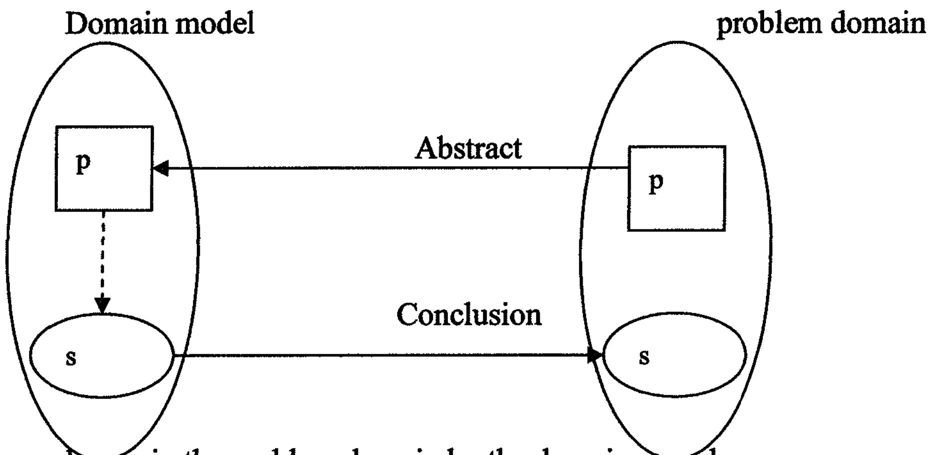

Fig.7 solving problems in the problem domain by the domain model

The basic requirement of knowledge representation is to make clear its own connotation, which is also called semantics. Each element in application should have its counterpart in the domain model. Usually, the basic features of knowledge representation are mainly categorized into four aspects:

- Sufficiency of knowledge representation: It shows that knowledge representation can cover all the knowledge needed in the problem domain.
- Sufficiency of reasoning: It shows that knowledge representation can cover all the reasoning processes and steps in the problem domain. The reasoning process generates new data structures through reasoning of the original ones. This is similar to the acquirement of new knowledge from the old knowledge.
- Efficiency of problem solving: It refers to the capacity to solve problems efficiently.

- Efficiency in learning: It describes the learning abilities of agents. An agent is able to get new information. It can integrate new information into its current knowledge structure. It can renew the current knowledge structure so as to better solve problems in the problem domain.

The approaches of knowledge representation involve predicate calculus, production rule, semantic web framework, etc.

The learning approaches can be used to construct the knowledge base. The development of the knowledge base can be divided into several phrases:

(l) Knowledge elicitation: Experts can add some knowledge which can be easily comprehended into the knowledge base. Some basic knowledge can also be obtained from the current knowledge base.

(2) Rule learning: Agents learn to solve some typical problems in application. They are also supposed to know the ways of solving problems. These ways are usually provided by experts. Through study and analysis of what it has got, an agent can generate some PVS rules which help solve similar problems.

(3) Rule refinement: Through experiments, examples, analysis and many other approaches, agents can further renew and refine the rules in the knowledge base. The process of rule refinement can be achieved by agents themselves. It can also be defined directly by experts.

(4) Exception handling: Mistakes and exceptions may occur. This is mainly due to the incompleteness of knowledge base, that is, the knowledge base may lack special examples or exceptions. In dealing with the exceptions, agents need more information. Or agents may require experts to provide such knowledge. Thus, the ability of agents in dealing with special problems is expanded. In this process, rules are improved and exceptions are eliminated.

4.2 Agent Communication Language (ACL)

Agent Communication Language (ACL), proposed by the Foundation for Intelligent Physical Agents (FIPA), is a proposed standard language for agent communications. Knowledge Query and Manipulation Language (KQML) is another proposed standard.

The most popular ACL are:

- FIPA-ACL (by the Foundation for Intelligent Physical Agents, a standardization consortium)
- KQML (Knowledge Query and Manipulation Language)[12]

4.2.1 Features of Agent communication language

When more than one agent is used in the multi-agent system, problems like intercommunication and collaboration may arise. In application, the intelligentization and the effective interaction between agents need further research. This is the same as that in the human society. Communications between human beings need not only the ability of using languages (such as English, Chinese, etc.), but the common understanding on knowledge in different fields. To software agents, effective interaction can not be realized without the following three basic components:

- The use of a common language;
- The common understanding of the exchanged knowledge;
- The ability to exchange what is included in the above two components.

Since an agent is a software entity which is intelligent and autonomic, it usually can be grouped into the knowledge level. Communications between agents should involve not only a character stream or binary stream, but also comprehension and communication. However, languages and protocols developed for other distributed computing environments can not provide effective support for an agent. This is because the language and protocols developed for distributed computing environments are used to deal with relations between processes. They are not for an agent program or the relations between these agent programs.

Therefore, it is very important to find an effective agent communication language which can support communications at a higher level. Otherwise, agents should have to take the trouble to do this from a higher level to a lower one or vice versa. As a result, an agent communication language (ACL) is developed. It is a protocol language used between agents to achieve the knowledge sharing and communication goal. It is also a crucial part of the multi-agent system.

Although agents and protocols are all used to deal with communications or problems concerning communications, ACL itself is not a protocol. Requirements of ACL are subdivided into seven broad categories. They are form, content, semantics, implementation, networking, environment and reliability. All these are further illustrated in the following seven categories:

1. Form

A good ACL should be syntactically simple, and easy to be understood by people. At the same time, it should be concise, easy to parse and to generate. To transmit a statement of the language to another agent, the statement must pass through the bit stream of the underlying transport mechanism. Thus, the language should be linear or should be easily translated into a linear form. Finally, because a communication language will be integrated into a wide variety of systems, its syntax should be extensible.

2. Content

ACL should be layered in a way that will be more suitable for the other systems. Particularly, a distinction must be made between the communication language and the content language. A communication language expresses communicative acts, while the content language is to express facts about the domain. Such layers can provide a conceptual framework for the understanding of the language. At the same time, it makes it easier to achieve the integration of the language to applications.

3. Semantics

The semantic description of communication languages and their primitives are usually limited to natural language descriptions. Nevertheless, it is still very important to use a formal description. This is mainly because the communication language is to be used in varied systems or environments. Different applications designers should have a shared understanding of the language, its primitives and the protocols associated. They should also abide by the rules in application. The semantics of a communication language should possess those properties possessed by any other language. It should be based on certain theory, and should not be ambiguous. In addition, a communication language should have canonical form, that is, similar meaning should have similar representation. Since a communication language is intended for interaction which extends over time amongst spatially dispersed applications, the semantics should pay special attention to time and location. Finally, the semantic description should provide a model of communication. And this model will provide a solid foundation for the performance modeling.

4. Implementation

The implementation of a communication language should be efficient and with a high speed. It should be able to fully utilize the bandwidth. It should possess the ability to cooperate with the existing software technology. The interface should be easy to use, and details of the networking layers should lie below the primitive communicative acts. Thus the user will be provided with a better communication environment. Finally, a communication language should allow partial implementation, because simple agents may only need to deal with part of the primitive communicative acts.

5. Networking

An agent communication language should accommodate well to advanced networking technology. It should support all the basic connections: point-to-point, broadcast and multicast. It should also support both synchronous and asynchronous connections. A communication language should have a rich enough set of primitives. In this way, higher-level languages and interaction protocols can be built based on these primitives. In addition, these higher-level protocols should be independent of the lower-level transport mechanisms used in application.

6. Environment

The working environment of an intelligent agent is usually distributed, heterogeneous and dynamic. Working in such an environment, an agent should build a communication channel to the outside world. Thus the communication language must provide tools for handling problems caused by heterogeneity and dynamism. It should support interoperability with other languages and protocols. It should also support knowledge discovery in large networks. Moreover, it should enable easy access to the existing systems.

7. Reliability

Communication among agents supported by a communication language should be reliable and secure. A communication language should also support private exchanges between two agents. Besides, it should provide a way to guarantee authentication of agents. Agents should not be assumed to be perfect. They should be robust to wrong messages. A communication language should provide reasonable mechanisms for identifying errors, and it should be able to send error reports and warnings to the user.

4.2.2 Knowledge Query Manipulation Language (KQML)

It is essential to design a common language for the interaction between agents. ACL is adopted by agents in their communication. ACL includes two main components: one is a Knowledge Interchange Format (KIF). It is an extension of first order logic. And it is used to represent the contents of messages. The other is Knowledge Query Manipulation Language (KQML). KQML is based on the Lisp language and plays a very more important part. KQML is the most widely used agent communication language at present. Its most important feature is that it includes all the information helpful in understanding the contents of messages. Standard communication primitives like "ask", "tell", "query", "delete", "insert" and protocol-oriented primitives like "subscribe" are all included in KQML. KQML also contains many protocols that support the interaction between agents. KQML predefines many actions. It predefines a sequence of actions that agents intend to communicate with each other. KQML can be regarded as consisting three layers: the content layer, the message layer and the communication layer (see Figure 8). The content layer mainly deals with the actual content of the message. It can use the Prolog code, C code, natural language descriptions or other representation language. The content layer is opaque to KQML. KQML can only be viewed as providing a package layer to wrap around the transport of content. It does not process the content of the message. The message layer is the essential part of KQML. The basic function of the message layer is to identify the protocol used for delivering the message. The function also includes the supply of a speech act which the sender attaches to the content. In addition, the message layer has some other optional features. It can describe the content layer in certain ways, such as the use of some language, terminology, or a descriptor naming a topic, etc. All these make it possible for KQML to analyze, route and deliver messages. [13]

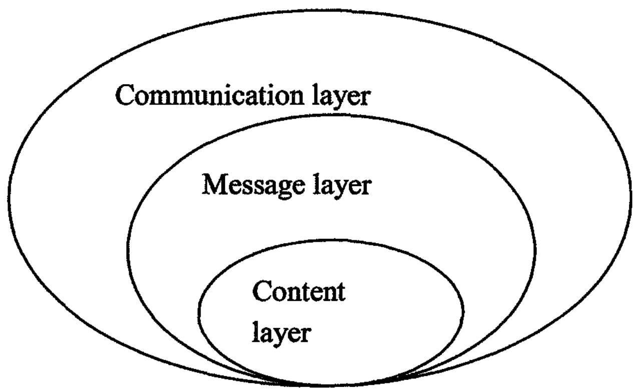

Figure.8 The three layers of KQML

The main features of KQML are as follows:

1. It emphasizes that an agent should be active, namely, it does not foretell whether an agent is used by a client or the server.
2. It provides some easy and effective language and does not limit the content of the communication.

The syntax of KQML is based on a balanced parenthesis list, which is similar to that of

LISP. The initial element is the performative, and the remaining elements can be regarded as the parameters. These parameters are known as keyword pairs. Here is a simple example:

(ask-one : content (PRICE IBM ?price)

: receiver stock-server

: language LPROLOG : ontology NYSE-TICKS)

Here, *ask-one* is the performative. It means that it is requiring for an answer. The parameter name *content* stands for the content of the message. The parameter *receiver* indicates the name of agent which will accept the requirement. The parameter *language* points out which language is used to represent the content. Along with other parameters (like *ontology*), the parameter *language* can help deal with the content of the message even when the content is not being understood (for example, they can help find a suitable agent which belongs to no specific group). The other function of these parameters is to help better understand the content of the communication: namely, the language and ontology used in the communication are quite helpful in understanding correctly the content of the communication.

The KQML language now has aroused much attention and has become the trend of development of the agent communication language. This is mainly because the KQML language better shows the three layers. It does not insist stubbornly on the unification of languages in the content of the communication. On the contrary, it supplies an effective and readable language, which enables the interaction between agents in different environment. Nevertheless, KQML has its own deficiencies. One big problem is that it does not provide a sound approach to deal with the relations between agents. Besides, KQML itself is not enough to support effective communication.

In a word, ACL is applied to the communication and collaboration between agents. It plays a basic role in the implementation of a multi-agent system. Some small-scale multi-agent systems may adopt certain simple protocols to achieve the goal. But, in the long run, formalization and standardization of ACL will help to large-scale agent software integration and interoperation.

4.3 The agent-based communication mechanism

The following communications mechanisms are in common use: direct communication, broadcast communication, the Federation System and the Blackboard System [7].

1. Direct communication is adopted when an agent knows exactly with whom it will communicate. And the two parties communicate directly with each other.
2. Broadcast communication means that broadcast messages are sent to all the agents registered in a specified group. If the broadcast message does not specify a group, the message is also sent to all the existing agents.

3. The features of the Federation System: The interaction between agents is based on the Federation Architecture. The Federation Architecture provides a collection of services to support the anonymous interaction between agents. Such services mainly include the following aspects: it can decide to accept and register an agent, record the capability of the agent and its task, provide communication services for the agent, respond to the requirements of the agent, and make it convenient for the knowledge interchange and the routing of information.

4 The Blackboard System is mainly based on the control module. The blackboard is a common area where agents post information that they can share.

4.4 Multi-agent coordination models

The basic requirement for a software entity to be an agent is autonomy, namely, its ability to interact autonomously with its environment. A multi-agent system includes many independent agents which can interact with each other, as well as with their environments. Therefore, to grasp the spirit of a multi-agent system, one should not only rely on the partial analysis of the internal architecture of each agent. It is more necessary to understand all the crucial factors from an overall point of view, such as the environment of the multi-agent system, the interaction between agents, the rules for coordination, and so on. There would be social behaviors when many people gather together to form social groups. It is the same with the multi-agent system. The interaction within the system will always present the complex features of the whole system. And these features usually can not be described or studied if just a single agent is taken into consideration. Hence the study of the coordination and control problem of the multi-agent system is of great significance. It can help better present the virtues of the system and regulate the operation of the whole system as well.

4.4.1 Definition of multi-agent coordination models

Generally speaking, coordination means the management of the relationship, such as dependency and interaction between two agents. In the field of computer programming, the most widely accepted definition of coordination is that coordination refers to the regulation of diverse elements into an integrated and harmonious operation [14].

In the multi-agent system, it is the interaction between agents that needs to be coordinated. The coordination models tend to provide a formal framework for the interaction between agents. Usually, it deals with the addition or deletion of agents, the communication activities between these agents, their distribution and mobility in the multi-agent system, the synchronization and distribution of the agent actions over time, and so on. To be more exact, a coordination model should include three elements, which are shown in Figure 9:

The coordinables: The coordinables are the entities in a multi-agent system. The interactions between them are regulated and controlled by the coordination model. The coordinables could be processes, thread, objects, etc. These coordinables could mainly be all kinds of agents in a multi-agent system.

The coordination media: The coordination media tends to give an abstract description of the coordination actions which make the interaction between agents possible. It is also the key for the coordinables to be organized together. For example, tuple spaces belong to natural coordination media.

Coordination laws: Coordination laws describe the actions by coordinable entities which help realize the coordination between agents. These laws can be defined on the basis of communication language or coordination language. The communication language presents the syntax which can express the data structure and carry out data interchange. The coordination language is a collection of both interaction primitive and semantics.

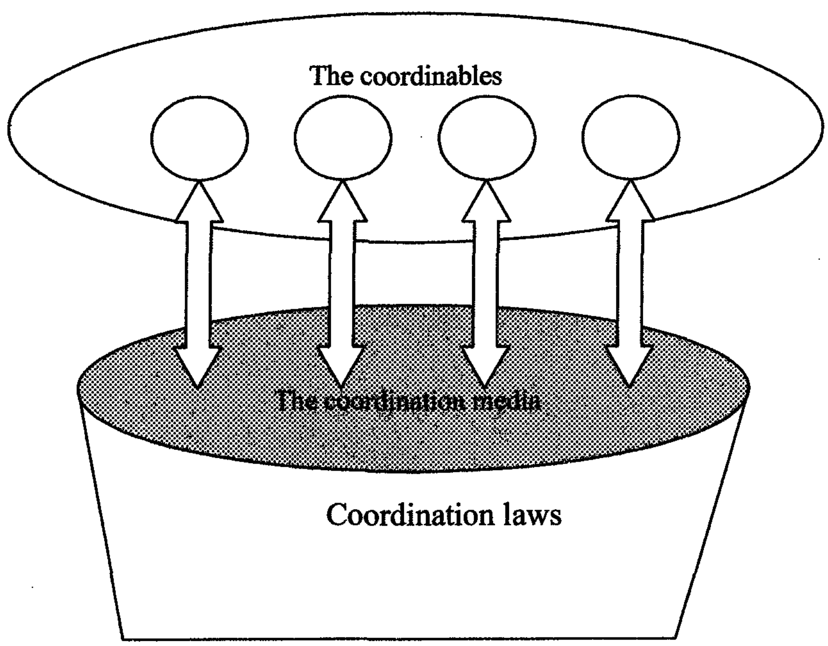

Fig. 9 The basic elements of a coordination model

From the software engineering point of view, the coordination model, as a basic mechanism, can effectively provide support the software system architecture and the development of multi-component software system. In the software engineering, the two terms which are closed related to "coordination" are configuration and architecture description. According to these two terms, the system is based on these components and the linkage among them. By distinguishing the behavioral description from the structural description of the components, the system tries to regulate the behavior and linkage of these components. There are similarities between the coordination model and the structure type. The latter is always used in the study of software structure. A coordination language embodies a coordination model. It is also regarded as a language for configuration and architecture description.

4.4.2 Classification of multi-agent coordination models

The coordination model and language can be classified from different aspects. For example, they can be classified into different kinds of coordinables, expansibility, openness, etc. Here is a simple and effective classifying approach, according to which the coordination model can be classified as control-driven and data-driven.

Control-driven coordination models: In the control-driven coordination models, the coordinables open themselves to the external world and interact with it through events. Look at the coordinables for the point of view of the coordination media. When the state changes, events will occur on these input/output ports of the coordinables. For example, an event could be announced by a process sending out data from an output port. Coordination laws describe how those events and states changes occur and how they progagate. The coordination media handle the interaction space among agents, paying no attention to the data actually exchanged between processes. It is only interested in the communication events. Figure 10 shows this model.

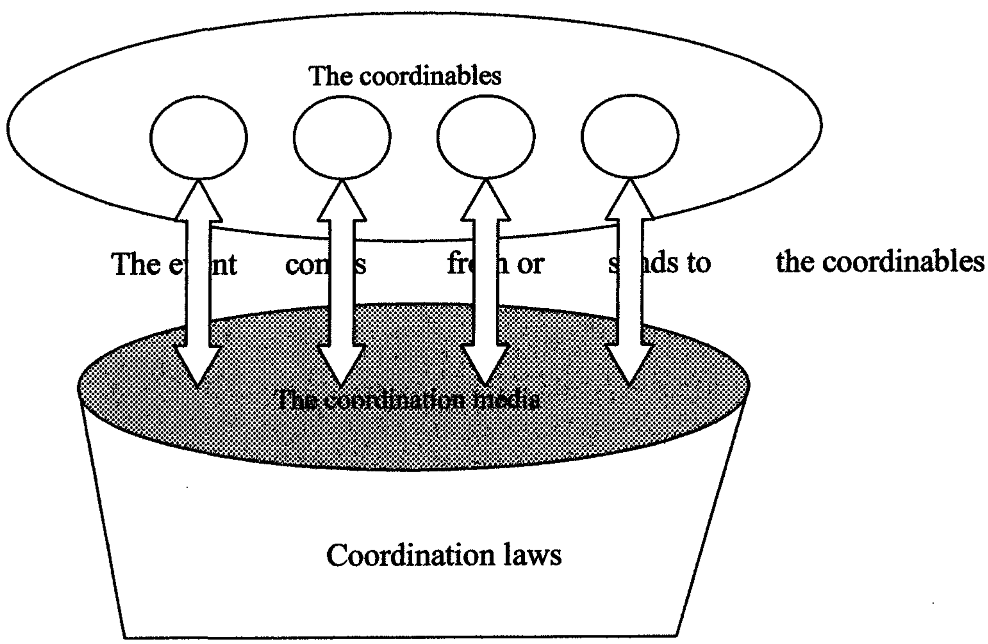

Fig.10 Control-driven coordination models

Data-driven coordination models: In data-driven coordination models, coordinables interact with the external world by exchanging data structure. The coordination media seems to act as a shared data space. Coordination laws establish the representation form of data structures and ways to store and extract data in the data space. Different from the control-driven coordination models, the coordination media pays no attention to the state changes of the coordinables. It does not provide any virtual connection among coordinables either. Figure 11 shows this model.

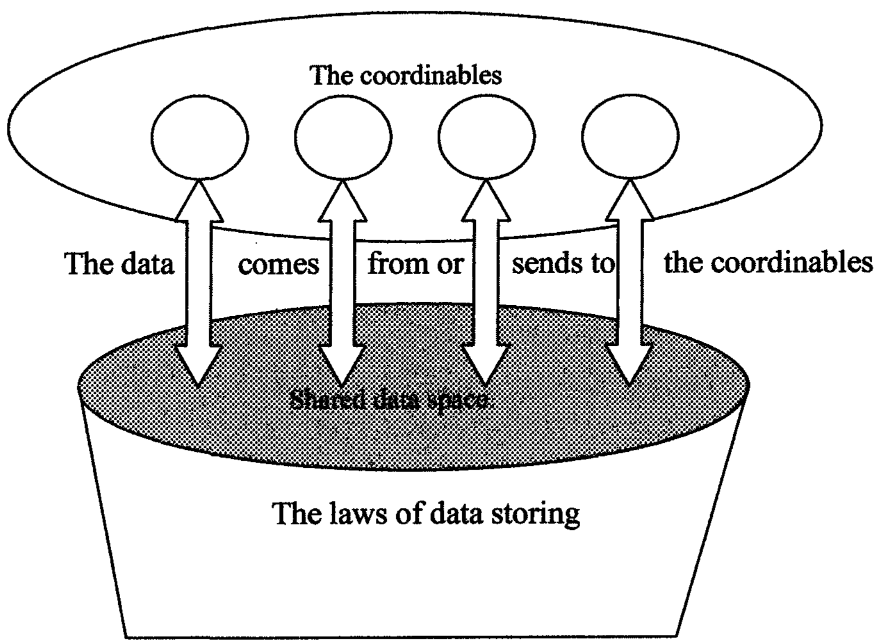

Fig.11 Data-driven coordination models

According to the format and syntax, control-driven coordination models can better support complete separation between computation and coordination activities. Such separation could be achieved through a new coordination language. In this coordination language, the computation enclosed in the coordinables can only be regarded as the black box with input and output. While in data-driven coordination models, the coordinated part is always a set of predefined primitives. These primitives are embedded in a major computing language. Though these primitives can encapsulate part of the communication and configuration, they have to be used with the help of data computation. This is because it is quite difficult to distinguish which part belongs to computation and which part coordination.

From the view-point of application, different coordination mechanism should be adopted in different applications. Therefore, the selection of a coordination model is very important to the design of the multi-agent system. Generally speaking, control-driven coordination models are more appropriate for those systems which have a clearly defined number of entities. The flow of control and dependencies existed in these systems need to be regulated. And at the same time, data exchange is not very important in these systems. For example, there are computational intensive parallel applications, distributed management systems, complex software architectures, etc. On the contrary, data-driven model seems to be more suitable for those open applications. In these applications, many autonomous entities which are also unknown to each other will cooperate. In this situation, it is very difficult to define and manage the dependencies between components. This is not the same as a control-driven model will do. Furthermore, this way of management may also clash with the autonomy of the components as well as the dynamicity of the open environment. Focusing on data can preserve the autonomy and dynamicity of agents. This is because autonomous agents usually get information rather than control from the external world.

4.5 Multi-agent negotiation models

In a multi-agent system, each agent acts on its motive and tries to realize its goal. An agent only possesses some imperfect knowledge of their environment. Hence negotiation becomes very important. The cooperation between agents will not occur naturally. There has to be a plan and it can only be realized through communication and negotiation. Negotiation includes the discussion which can communicate messages. Discussion intends to produce an agreement. It is a repeated process where agents can exchange information, interact with other and finally reach an agreement. Thus, it is an important mechanism to achieve cooperation and agreement between agents. An agent can do this through different ways. At the same time, agents have to evaluate the information provided by other agents before they decide finally on the changes of their intentions and behaviors.

To negotiate more effectively, agents have to possess the following three abilities:

1. Agents should be able to represent and maintain its beliefs, desires, goals and intentions;

2. Agents should be able to reasoning about the other agents' beliefs, desires, goals and intentions;
3. Agents should be able to influence other agents' beliefs, intentions and behaviors.

When each agent in a multi-agent system decides its actions to achieve its own goal with the help of discussion, the discussion between agents does not belong to negotiation. Discussion then can be regarded as a process where messages are mutually exchanged to reduce clashes and promote the accomplishment of the goal. Agents should use some information which can dynamically change the behaviors of other agents, so as to build up some knowledge. There is an analogy between such knowledge and the credibility between people in a human society. Such knowledge can be used to evaluate the information agents provided. Agents will renew their information according to the responses of other agents. Thus they can further classify and improve the knowledge available.

Negotiation between agents needs to follow certain protocols and strategies.

Multi-Agent Interaction Protocols represent abstract and formal patterns of agent interaction. It reflects the rules and purpose of interaction between agents. It is also closely related with the inference mechanism and thus is the focus of the multi-agent system.

Multi-Agent Interaction strategies mainly concern the following aspects: task analysis, finding relevant information of other agents, and analysis of the relevant interaction protocols. At the current moment, most researches adopt the multi-agent system. But it is the designers who will analyze and decide the interaction strategies beforehand in the designing process. However, it is more important for agents to choose interaction strategies autonomously, which is also regarded as the development of the dynamic interaction strategies. At present, research is not yet mature in this field and needs to be further developed.

4.6 Task decomposing and scheduling

When one task needs to be solved by a set of agents, the first problem is which agent undertakes which part of the task and at what time. To distribute a task, formal description and decomposing of the task can not be avoided.

The description of the task requires defining the scope of the problem, and identifying the known and unknown conditions, so as to express the problem formally. Actor Model is the initial research result. And later Davis and Smith put forward the contract net protocol. But nobody has ever mentioned the autonomous description of the task. Usually the task description in Contract Net is manually performed by the designer. Task decomposing is to decompose the task into the sub-tasks which need less knowledge and resource to be accomplished. The task usually is fulfilled by some particular agents. Task decomposing needs to take into consideration the requirements of the task, the manner of decomposing of the task, the agents and resources involved, and the relationship between sub-tasks.[15]

According to the different features, such as space, time, logic or function, the task can be decomposed into sub-tasks in different ways. Task decomposing mainly depends on the following features:

- Abstractness: Sub-tasks at different levels of abstraction are achieved respectively by different hierarchies of control in the hierarchical system.
- Coordination: In task decomposing, the performer of the sub-tasks should be able to control the whole process. And the coordinated relationship should be clear and simple. Autonomy: Sub-tasks are less dependent on each other.
- Data dependence: Tasks which present a close relationship between knowledge and data are grouped together.
- Function/Product group: Tasks that will accomplish certain function or produce certain product are grouped together.
- Interaction: The number of sub-tasks should not be too large. Otherwise it will result in too much interaction and the complication of the media.
- Redundancy: Division of the redundant sub-tasks will improve the validity and decrease the uncertainty.
- Decrease in resource demand: The division of tasks should be based on the principle of decrease in resource demand.

Though the theories, approaches and validated technologies concerning the task decomposing of the DAI system have not been reported up to now, researchers have put forward several approaches for the task decomposing.

The following are the approaches usually mentioned:[16]

1. Decomposing according to the intrinsic features of the task representation: The task representation, such as the state, state space, performer and data structure, implies the approaches for task decomposing based on the intrinsic features.
2. Hierarchical specification: Decomposing the task into sub-tasks should be based on the decomposing goal.
3. The load balance approach: Decompose a large number of tasks into sub-tasks in terms of optimal load balancing, such as communication and computation.
4. The least related subgraph: If a graph structure can be used to represent the relations between different components, we can adopt an algorithm to get the subgraphs, namely, the sub-tasks
5. Subtask combination: A big task will be decomposed into sub-tasks in accordance with the collaboration between agents. And each agent will solve part of the problem. Specified tasks are assigned to different agents to confirm their responsibilities for the actions. The assignment of tasks to agents is elementary. And the tasks can be manually distributed to agents according to the predefined

function of each agent. However, the research of DAI pays more attention to the dynamic assignment among agents through certain techniques. In 1990, Yoav Shoham and some others brought forward Agent-Oriented Programming (AOP).

Presently, researchers have studied the task allocation of the basis of the following principles: [16]

1. To avoid the bottleneck: In the task allocation, the key point is to avoid too many resources being involved.
2. To select the most appropriate agents: Each particular task will be allotted to the most appropriate agent.
3. To use the relevant knowledge: The allotment of tasks is restricted by the task sequence. In addition, the agent, who has the strongest ability to interact with the other agents, will undertake the coordination.
4. To perform complex roles: Each agent can take several roles to realize its flexibility and consistency.
5. To redistribute redundancy: Those uncertain tasks will be redistributed to decrease the uncertainty and improve the validity.
6. To consume the least resources: The distribution of tasks will be based on the least consumption of the key resources.

CHAPTER 5

Multi-Agent System (MAS) model

A multi-agent system (MAS) is a system composed of multiple interacting intelligent agents. Multi-agent systems can be used to solve problems which are difficult or impossible for an individual agent or monolithic system to solve. Examples of problems which are appropriate to multi-agent systems research include online trading, [17] disaster response, [18] and model social structures. [19]

5.1 Model of MAS used in application system

1. The application system can be regarded as the grouping of certain agents. The behavior and function of the application system depend on the interaction between agents.

2. The application system can provide special services to different users through the shared environment.

3. Independent application programs can interact with each other.

5.2 The architecture based on the model

5.2.1 Architectural structure

The architectural structure includes the following components. Figure 12 shows the architectural structure:

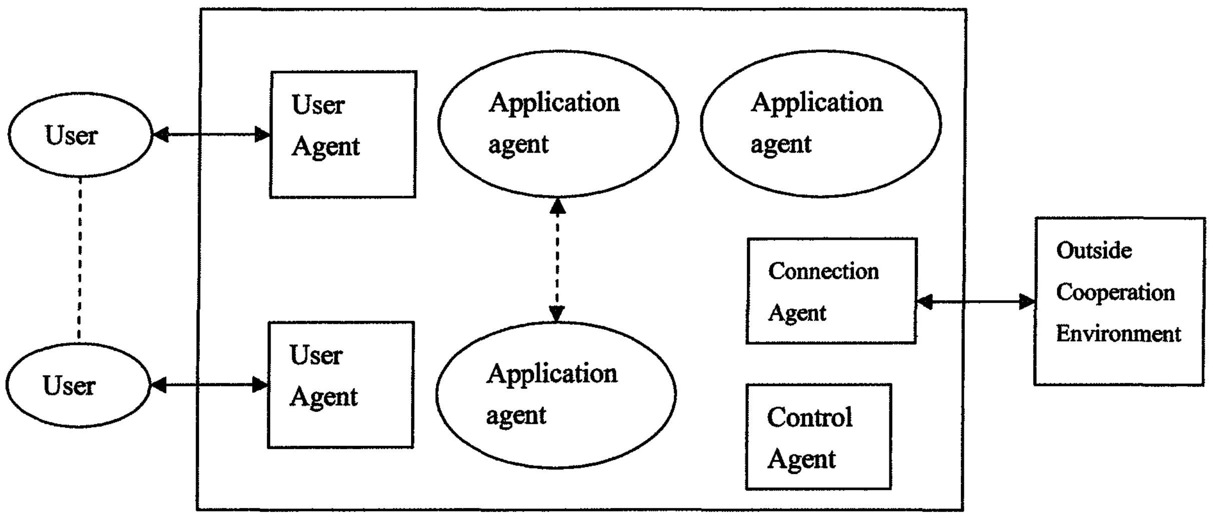

Fig.12 The architectural structure

User interface: It is the interface of the application system. In reality, it is displayed on the user's web browser. It can be sent to other agents according to the user's requirement.

User agent: It represents the user in a model. At certain moment, the user himself may or may not appear in the shared working space. So the user agent will accept and send the relevant information to other agents. Furthermore, the user agent will achieve some tasks on behalf of the user according to the user's interests.

Application agent: It is the application agent that performs the application task. Each application agent has its own special task. It begins or stops functioning in terms of the requirements. In the system, application agents can interact with and provide services for each other. In this way, they can help realize the functions of the system.

Outer agent: It provides quick access to the MAS. Thus it is easier for the MAS to interact with the outside world. If the application system needs another environment, network service or middleware platform, it can ask for help from certain outer agent. Each outer agent deals with one particular environment. At the same time, it translates or transfers the service request and sends back the answer to the corresponding application agent.

Control agent: It provides basic services to the application agent. It controls all the agents, including the startup and cessation of an agent.

5.2.2 User interface design

The user request and its relation with the user agent result in the user interface. The user interface mainly has the following functions:

1. Showing the interface in use;
2. Showing the messages kept in the user agent;

3. Permitting the user to interact with another one and starting up its own user agent in the system.

The user may interact with the user agent through the user interface displayed on the user's web browser. When the user sends service request to the web browser, the user interface comes into being. It is generated by the user agent. Usually the user agent is indicated by URL. When it contacts with the user agent for the first time, it needs a log-in process. User identity and password are needed.

The user interface design covers three parts: the first part includes the messages and listing. The user can choose one message out of the listing. The second part refers to the control area which controls the behaviors of the user agent and the user interface. The third part is the display area where many messages are displayed. The display area can also be regarded as an interactive region.

The user interacts with the user agent through the user interface. The process of interaction consists of the following parts:

1. The user selects one message from the listing in the control area, and asks the user agent to perform the message processing.

2. Through HTTP, the user agent gets the relevant data and the initial graphical user interface (GUI), which are all stored in certain browser cache as the static HTML file. This GUI is stored as the user data.

3. The user selects the message from the queue and controls the initial interface, on which some control tools are designed to interact with application.

4. The user uses the control tools on the user interface. This can cause the user agent to adopt certain method used by the application agent. The user agent issues a service request according to the provided application interface, and sends the request. The result of the interaction with the application agent is that a character string containing a new application interface is generated and stored as the user data.

5. The message is displayed in the form of a new GUI, in which the result of the last operation is included. So the interaction between the user and application is based on the transfer of GUI from the application agent to user agent. It is stored in the message queue as a message, and transferred to the display area in the user interface. Then it is displayed for further interaction between the user and application.

5.2.3 User agent design

The structure of the user agent contains three parts:

1: The interface for interaction with the user, which can be seen outside.

2: The interface for interaction with the system, which is the same with the application interface.

3: The LIB for storing the results of services.

The structure of the user agent is displayed in Figure 13.

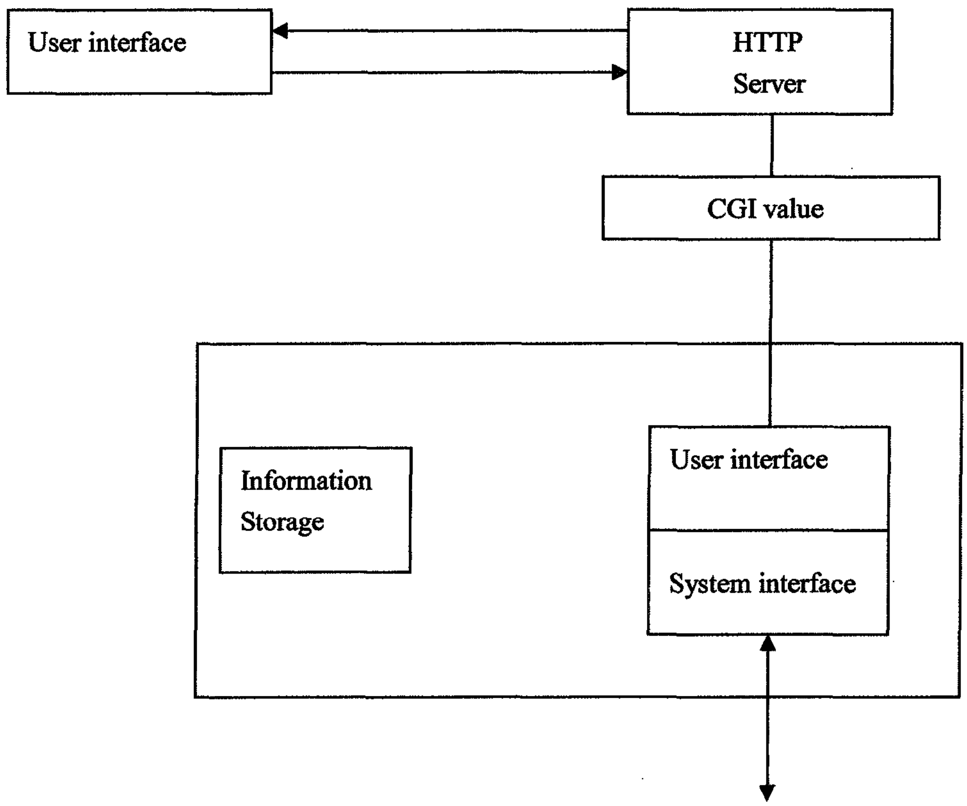

Figure.13 The structure of the user agent

The three parts can be integrated into JAVAVM in the form of a HTTP server thread, database and user agent thread. There are several other points to be considered:

The user's URL is linked to a CGIscript.

CGIscript connects the user agent server which is in the same machine with the HTTP server. It sends to the server the environmental variables defined in CGI and the variables transferred by the query string of URL.

The server of the user agent receives the parameters through Socket. And the HTML text generated by the user agent is sent back through Socket. This standard output is transferred to the browser by HTTP server.

In a user agent, the message queue and the information of the user should be stored in one database. Therefore, a database should be built.

5.2.4 Application agent design

In this model, the application agent can provide services initiatively. Without the application agent, it is impossible to build up the whole system.

We could take into account the following aspects:

1: The application agent should be independent of the operating system and the physical machine;

2: What the application agent offers can form a platform, and thus programmers need not concern themselves with the preliminary implementation;

3: The application agent should support the unification operation based on data and services;

4: The application agent should be fault-tolerant when some links or data are destroyed.

To achieve this goal, we should use JAVA as the program language. It is supported by most of the computer systems. At the same time, the object-oriented approach allows programmers to use some classes without considering the implementation.

An application agent contains three parts: the kernel, the service control, and the external linkage. They have the following functions.

The kernel is the main program that stays in memory at all times to perform all basic operating system functions. It is provided as the control module. It can support the model (the service control). The model contains service request and service provision. It also manages the module connected with the outer space (outer connection). The kernel controls them through checking the thread status continuously. If there are errors in the service control or external linkage, or it is time to terminate the program according to other factors, the status of the kernel has to be changed.

The external linkage is one thread in the kernel. It links the application agent and the relevant control agent together. The external linkage has three tasks to perform:

1. It should be able to search for messages in the intercommunicating system (the internal library). The messages should be transferred from the service control to the control agent linked.

2. It should be able to search for messages in external communicating system (socket). The message is transferred from the control agent to the service control for collaboration.

3. It should be able to check its linkage with the control agent constantly so as to find out the disconnection or the termination of the control agent.

The service control can be regarded as a thread. It functions as the application agent. It performs the services for the application agent. The service control must register its startup and termination in the kernel.

The structure of the application agent is presented in Figure 14.

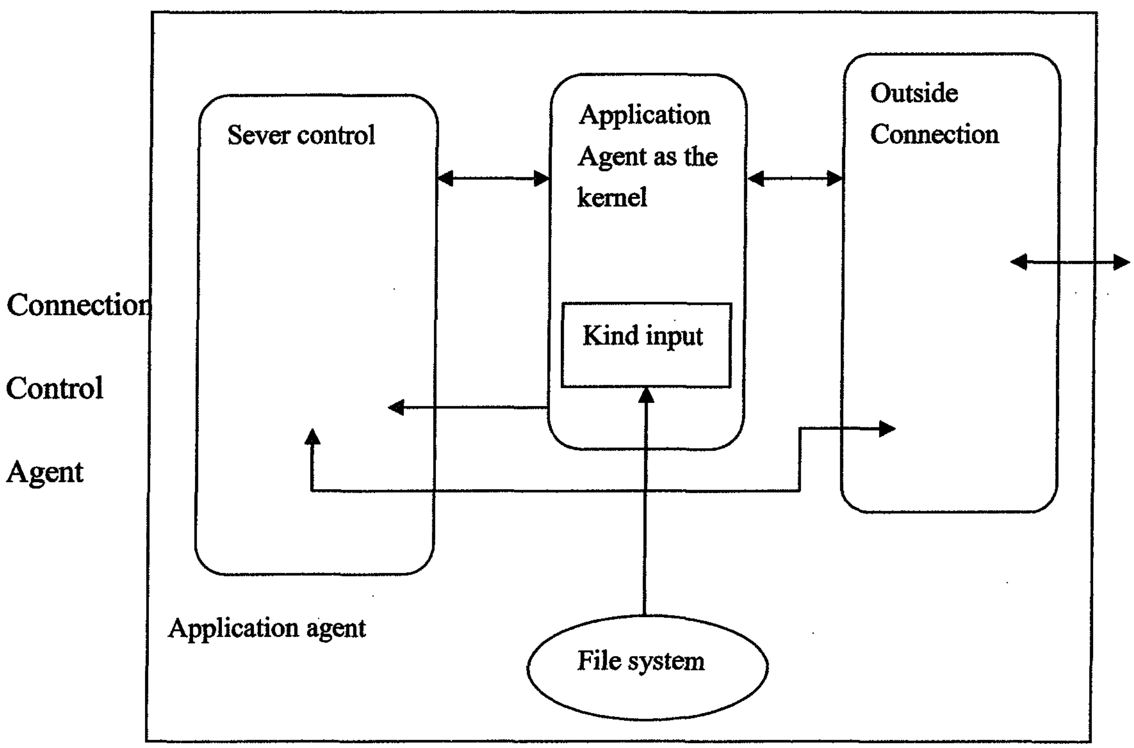

Figure.14 The structure of the application agent

5.2.5 Development of the Control Agent

The control agent can be regarded as the control module in a multi-agent system. It causes the appearance of the virtual space of the multi-agent system where agents can interact with each other. Application agents, user agents and external agents can only be accepted in the system when they are linked with the control agent. From this point of view, the control agent can be regarded as a way leading to the collaboration between agents. The functions of the control agent can be divided into two major categories:

1 : The control agent can realize the communication and collaboration between agents. It has its shared database, which can be used to store information by all agents in the system. Since the multi-agent system is a distributed computing platform, it can not adopt the centralized architecture. The control agent must have the shared virtual space in which agents can communicate with each other.

2 : The control agent manages all the application agents, user agents and external agents linked. It can deal with the register of an agent into the system and acquire its state. It is also responsible for its linkage with another control agent. Through many a linkage between control agents and the incorporation of new control agents, the whole multi-agent system can be expanded.

The ability of fault-tolerance of the whole system depends on the interactive protocol between agents as well as the architecture of the system. That is to say, it depends on the linkage between control agents. Thus the control agent should support the reliability of the entire system and achieve the following goals:

1: Failure of part of the control agents will stop the whole agent;

2: The modules which are used to realize the linkage between control agents should be easily adaptable. Only in this way, it can improve the reliability of the system and can be applied into some special network structure;

3: The control agent is able to reenter into the system, so that the faults made can be corrected when the system restarts.

Control Agent Architecture

Architectural structure

The design principle of the control agent is to adopt modules. Each module interacts with the other parts of the agent through data transmitted with the help of the communication equipment. Like application agent, each module uses the internal database. Since the control agent should be able to reenter into the system and renew when it is terminated, it is of great importance to have a copy in the modules. Each module should be able to copy itself, keep and restore the relevant information. Accordingly, the control agent can consist of the connecting module, library module and the kernel.

The kernel is the essential part of each control agent. It uses the main program to call other modules so as to control them through constant checking of their states. If a mistake occurs in one module, the kernel will terminate the other modules in order to keep the relevant information. In this way, the kernel can ensure the termination of the control of other agents when one of the modules makes a mistake.

The most important part of the control agent is the kernel. At the very beginning, it calls the other modules. Then it manages the thread generated and checks the mistakes that have caused the termination of the control agent. At the startup stage, the kernel mainly executes the initialization function, such as locating command line options, analyzing the files which provide information for the current multi-agent system, etc. When these tasks are achieved, the kernel will perform the function of the control agent. The difference between the startup and operational phase of the kernel is only due to their various aims, instead of the different functions of the kernel.

User agent server treats the user agent as part of the control. The user agent consists of the following aspects: the interface interacting with the user, the interface interacting with the multi-agent system, and the library receiving and storing information. All the functions of the user agent will be included in the control agent.

The control agent connector establishes and manages the linkage with other control agents. This module helps realize the communication protocol of the system. The control agent connector is the interface for other control agents in the system.

The cooperator base provides services. The cooperator base is formed by the buffer area which is used to save the tuple.

Application agent connector provides an approach to the collaboration between application agents and external agents. On the other hand, application agent connector links together these agents, saving information about when they start or terminate. The external agent is a special kind of application agent. It has the interface through which it can be linked with other collaboration environments.

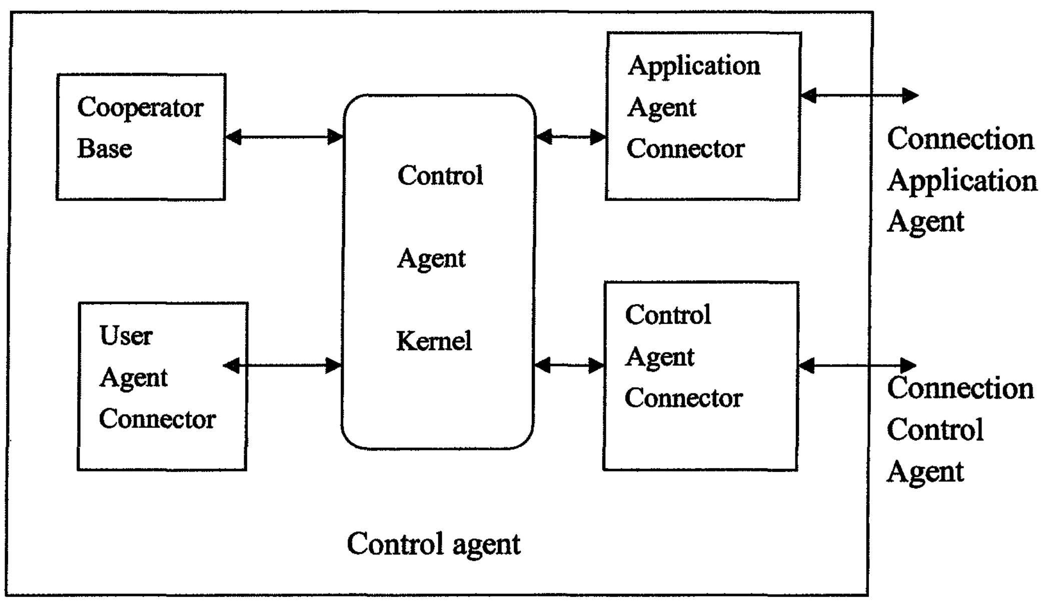

Figure.15 The structure of Control Agent

CHAPTER 6

The application example of Multi-Agent System — Intelligent Tutoring System on Network

6.1 Overview

Reasons for adopting the agent technology to solve problems:

Taking the features of the agent and the whole system into consideration, we have the following reasons for adopting the agent technology to solve problems in network teaching.

1. In network teaching, data distribution, problem solving and reactivity should be consistent with the distributivity, encapsulation and modularization in agent technology.
2. Problems like coordination and information sharing may result in a complex relationship between application parts. This calls for an agent which has the social ability to solve the problems.
3. Distributed object system has the ability of encapsulation. But it does not possess the ability of complex reasoning needed by interreaction and autonomy. Ordinary distributed information processing systems can resolve problems caused by distribution. However, they can not show the autonomy of the elements.

Therefore, according to the factors mentioned above, the most effective way is to adopt the agent technology to achieve a better network teaching result.

6.2 CAI Modeling

Computer assisted instruction (CAI) platform provides a global support for collaborative learning in networking teaching. It performs many functions, such as measuring the learning styles, grouping, structuring the curriculum, learning, communicating and discussing with each other, assessing the learning results, answering questions, assessing the collaborative performance, and so on. Learners can thus fully utilize their subjective initiative in the approved environment created by the CAI platform. Accordingly, they can construct their knowledge systems. Generally speaking, ordinary network teaching can provide a good support for the collaborative learning. However, it lacks intellectuality in some degree. After class, learners' collaborative learning still needs the help of a tutor. The tutor can help in many different ways through hand operation. For example, the tutor can coordinate the communication among learners, encourage them to speak, inform the time and place of a meeting, answer questions, etc. On the one hand, this is a waste of human resource. On the other hand, the involvement of people in networking teaching will result in belated feedback, wrong information, and other problems. Therefore, it is quite necessary to develop an intelligent electronic tutor to replace a real person, so as to better achieve the tasks.

6.2.1 Components of the System

The CAI system consists of the user agent, application agent and control agent.

6.2.1.1 The user agent

Learner agent: The learner agent acts as the interface between learners and the whole teaching system. It provides a humanized interactive interface. It can help achieve the interaction between the learner and the entire teaching system while taking the special characters of each learner (such as the different learning content, different timetables and even the various learning styles) into consideration. In addition, it shares these characters of learners with other agents.

Instructor agent: The instructor agent acts as the interface between the instructor and the whole system. According to this agent, the instructor or so-called tutors can interact well with the whole teaching system. At the same time, it can instruct and monitor learners during the teaching process. The agent can actively acquire the learning process of each learner from the database which stores the various characters of learners. It can adjust and intervene the learning process through certain learner agent or the database which allows searching within different specialities.

6.2.1.2 The application agent

Tutoring agent: The tutoring agent mainly helps instructors monitor the learning process of different learners. It provides resources for learners and answers questions during the collaborative learning process. Besides, it assesses the learning results and responds to learners' assignments. To put it more specifically, the tutoring agent has the following functions:

1. Grouping learners according to their different characters.

What collaborative learning emphasizes is that students should cooperate with each other in their learning when they are in groups. This has a beneficial impact on learning. So at the very beginning, all the learners are grouped. And research shows that heterogeneous teams can usually carry out effectively the collaborative learning method and perform complex tasks required. Therefore, how to group learners becomes quite essential. It is one of the most important tasks of an e-tutor.

2. Giving instructions to learners in collaborative learning groups.

With the network teaching platform, learners carry out the collaborative learning process mostly without the presence of an instructor. Learners may get confused in their self-study, and might come across all kinds of questions. The tutor agent will act as the instructor and give certain instructions to them. Thus better learning results will be obtained.

3. Discussion.

The tutoring agent provides a locality for learners' collaboration and discussion. It records autonomously what has been discussed. It adjusts the teaching sources and the teaching process according to those questions frequently asked by learners.

4. Structuring the curriculum.

Make the learning content be consistent with the characters of the learner. The relevant courses are obtained from the resource agent. At the same time, the characters of the learner and his or her inclination will all be taken into consideration. Or try to get a curriculum framework from the resource agent. With the teaching materials and relevant information acquired from the resource agent (which includes the total resources and individual resources), we can get appropriate teaching courses (or teaching procedures). Thus in the teaching process, the teaching materials and styles can be adjusted according to the learning progress and learning results. Furthermore, it also plays an important part in resetting the time for learners so that they can finish the entire course within a maximum period of time.

5. Distributing assignments

The instructor will allot some tasks to learners over a period of time. Learners are asked to perform the tasks according to certain requirements. The e-tutor can inform learners the content of the task, the requirements, the deadline and other information as well.

6. Publicizing distance classes and meetings.

Some of the topics need to be discussed by the whole class together. Some need the instructor design activities for all learners. The e-tutor will beforehand inform learners the time, location and topic of the meeting or the class.

7. Checking periodically the learning results of each collaborative group.

It mainly checks what has been discussed, who has participated in the discussion, and how learners have performed their tasks. It will caution learners who have not finished their assignment before the deadline and inquire the reasons. Thus it can urge them to make progress. It also praises members who have actively participated in the discussion.

8. Assessing the learning results.

Tests generated automatically can assess the learning level of each learner. At the beginning, it will assess the entry skills before learning. During the learning process, it can know in time the progress a learner made. This is based on what he or she has studied and according to the formative assessment. When the learner finished a phrase of study, it can provide an objective summative evaluation. The assessment of the entry skills before learning and the formative assessment both aim at building a better teaching mode and improving the teaching materials. Thus it has a strong purpose. And the summative evaluation has its own objective criteria. It is a fair evaluation for any learner.

9. Answering questions.

Its major function is to provide answers to certain questions. It requires higher intellectualization. It can provide comparatively satisfying answers with the help of other agents. To those questions it can not find an answer, it will ask for help from the instructor.

Resource agent: The resource agent is the database covering different specialities. It can also collect information actively. It is able to provide maximum resource information concerning a speciality for each learner. The agent can choose from the existing courses or curriculum frameworks. There are teaching processes, teaching methods, teaching steps to be chosen by the agent. But there are not concrete teaching materials for the agent. According to the intervention of the instructor and the learners' response, the agent can actively adjust and expand its database. It can obtain relevant information from the Web and regroup it to make it become more useful.

Management agent: The management agent mainly exercises macrocontrol of the whole teaching process. It can actively get data of other agents and generate automatically management data. These management data include the time and locations for classes, various levels of learners, the tasks of the instructor, etc. It can help facilitate rapid response.

Outer agent: The outer agent links the entire multi-agent system with the outside environment. If the application system needs to interact with other environments, network services or middleware, it can ask the outer agent to provide such services. For example, learners may need information from the Internet, and the outer agent thus will help search such information.

6.2.1.3 The Control Agent

Control Agent: The user agent (learner agent and instructor agent) and the outer agent can only be used in the system when they are linked with the control agent. Thus, the control agent can be regarded as the means to realize the cooperation between agents. Besides, the control agent is responsible for coordinating relations among agents. It also supports the communication and collaboration among agents. It can help resolve problems when clashes or other conflicts occur among diverse agents. The other important function of the control agent is that it can keep in touch with the instructor, and accordingly the instructor can know well the learning process of learners.

6.2.2 System structure

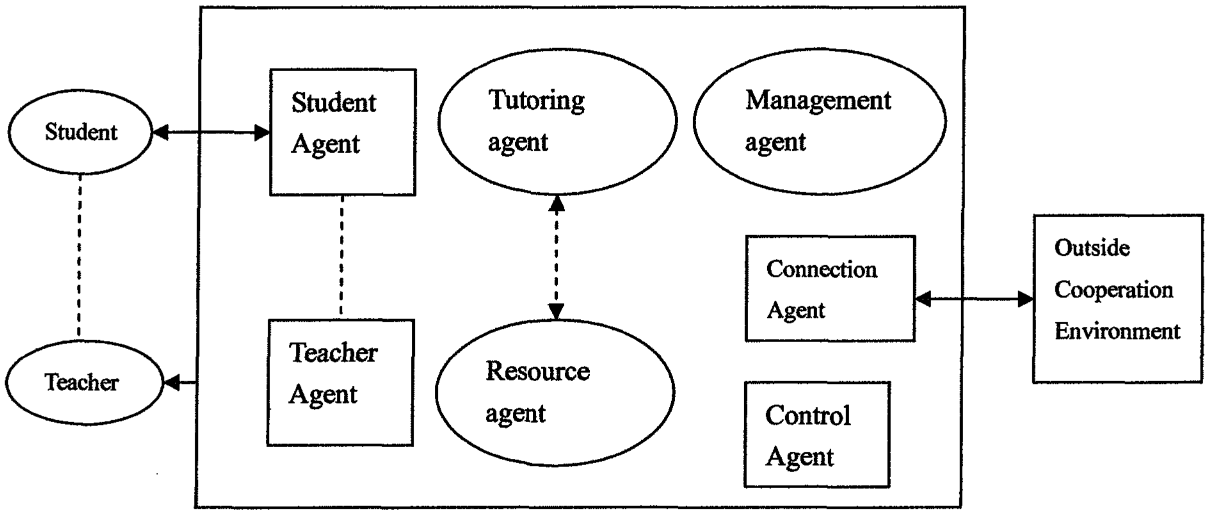

Figure.16 The structure of the network tutoring system

6.2.3 Tutoring Agent

6.2.3.1 The tutoring agent and the data flow

According to the functions of the e-tutor in network teaching, the tutoring agent consists of three modules. This is designed in terms of the structure of the computer system. These three modules are behavior-captured agent (BCA), behavior processing agent (BPA) and information feedback agent (IFA) respectively.

- The behavior-captured agent is responsible for the (instructor's or learners') behavior. It analyzes and preprocesses the behavior and translates it into a form which can be understood by the behavior-processing agent. Then the behavior-processing agent can deal with it. This function is similar to that of the computer input devices, such as the keyboard, the mouse, etc.

- The behavior-processing agent takes charge of the connection with the knowledge database. It searches for approaches to deals with the preprocessed behavior. Then the information feedback agent will react to the user or the system. With the system trigger, the behavior-processing agent can also automatically interact with the information feedback agent and serve the learners. Since different learners use different computers, their communication is mainly realized through the behavior-processing agent. And in this way, they will perform collaborative learning activities effectively. This function is the same with that of the central processing unit (CPU).
- The information feedback agent naturally is in charge of the interaction with the user (the instructor or learners) or the system. Based on the instructions of the behavior-processing agent, it can either react to the user's input or search the knowledge data base and update it. This function is similar to that of the computer input devices, such as the display, the hard disk, etc. Figure 17 shows the tutoring agent and the data-flow process.

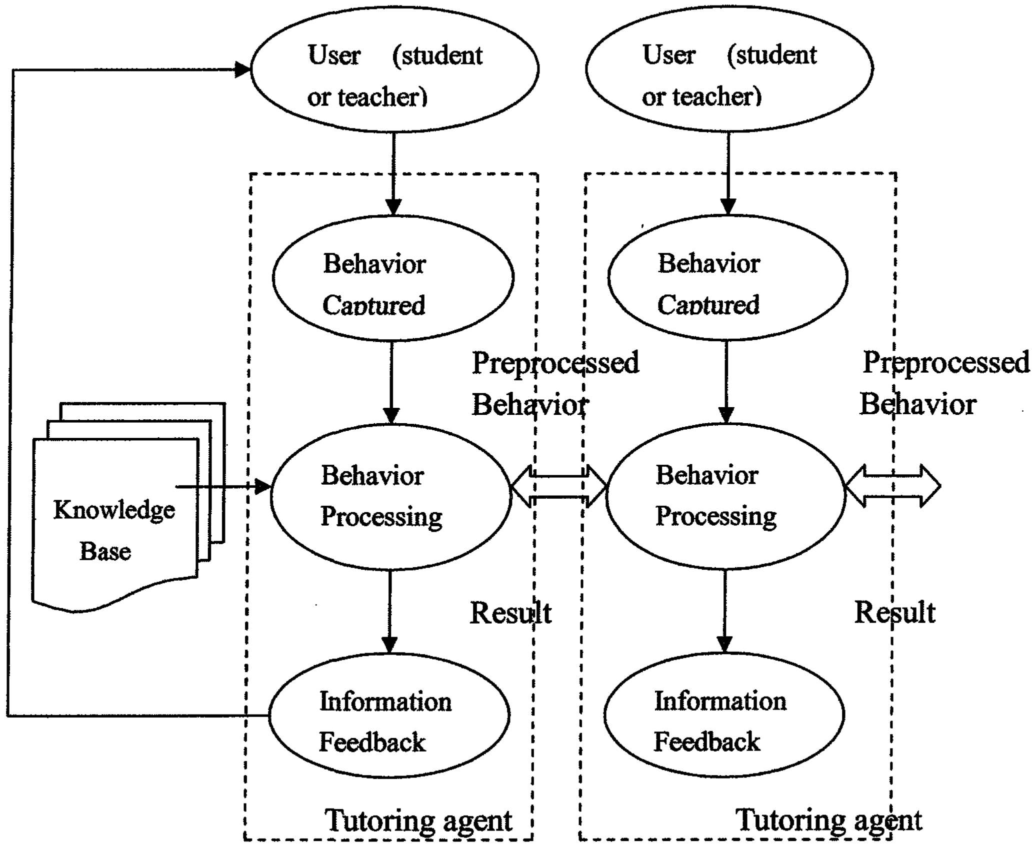

Figure.17 The structure and data-flow process of the tutoring agent

6.2.4 The System Database

To realize the functions of the e-tutor in network teaching, the server should have the following databases:

(l) Database used for selecting behaviors

This database is mainly used to select the behaviors of the user (the instructor or learners) and analyze them.

It records the valid operation of the user and then presents its characters (, which is also called the preprocessed behavior). For example, when a learner rapidly clicks on random links in the system, the behavior-captured agent will record the learner's behavior and compare it with those behaviors stored in the database. Then it will find that the learner is looking for some sources and he or she is not familiar with the networking teaching platform. After the behavior-captured agent has preprocessed the behavior, it will inform the behavior-processing agent. The latter will automatically process it. In addition, the information feedback agent will ask the learner what he or she is looking for. It can actively provide links to online resources to support the learning.

(2) Database used for processing behaviors

This database stores behaviors concerning the special characters of the user and the approaches to be adopted by the system. As we have mentioned, when the user is not familiar with the network, the behavior-processing agent can automatically search the database and help the learner find the relevant websites.

(3) Database used for storing information about the construction of network teaching platform

This database mainly stores information about the construction of network teaching platform. It also stores links to relevant websites.

(4) The questions and answers database

Experts check the questions and answers database for accuracy. It contains previously prepared questions and answers. During the collaborative learning process, new questions asked by learners and answers based on the discussion between the instructor and learners are also included in the database. Without any doubt, this database is mainly used for answering questions or providing model answers for exams.

(5) Instructional strategy database

The instructional strategy database mainly stores the instructional strategies. During the collaborative learning process, most of the learners will work in teams. However, instructions from the instructor are unavoidable. The instructional strategy database is able to help the system give instructions to learners.

(6) Collaborative strategy database

This database stores information about the grouping of learners. In network teaching, grouping means that learners are divided into groups according to the collaborative strategies and learning resources.

(7) Archival database concerning learners' information

Grouping of the learners always involves personal information. Therefore, it is necessary to build a database to help perform the grouping function.

(8) Database used for storing learners' characters

This database stores learners' characters like the cognitive ability, the existing level of their knowledge, and the learning styles. When learners are divided into different groups, these characters should be taken into consideration.

(9) Material database

The material database contains all the materials needed in class, which is also called script used in class. It includes previously prepared questions, figures, graphs, teaching materials, prompting messages, and feedback information indicating whether or not the learners' responses are correct, etc.

6.3 Network tutoring process instance

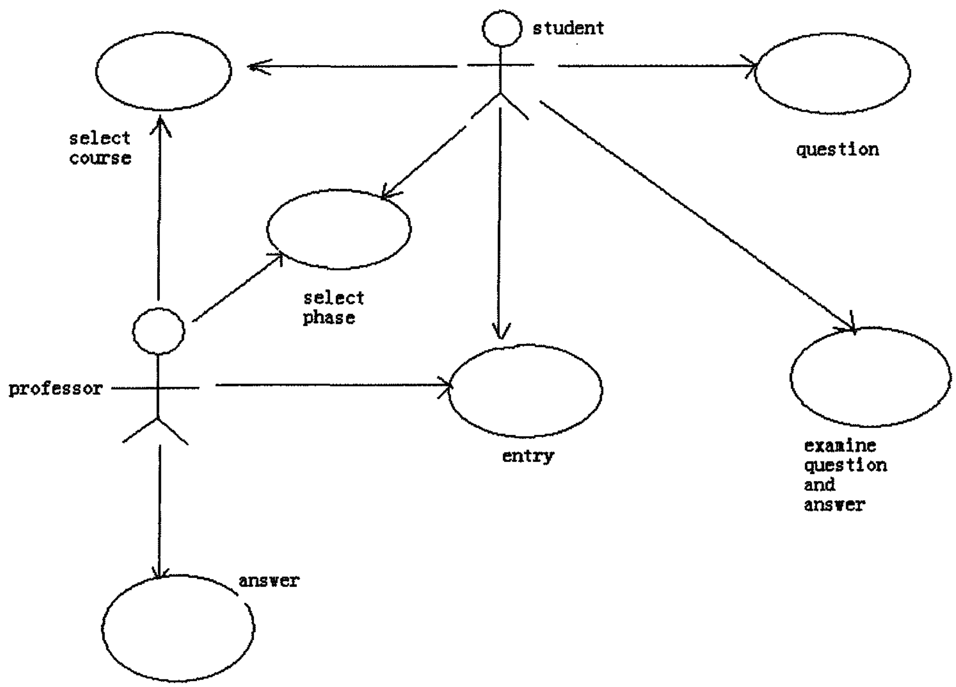

Fig.18 Network tutoring system instance

6.3.1 User agent-student agent

entry(identity validate)——courses select——phases select(preparation, discuss, question, exercise, evaluate)——practice—refer

Preparation: Tutoring Agent gives the relevant material based on the student's instance;

Discuss: Tutoring Agent distributes the discussing section based on the student's instance;

Question: Tutoring Agent gives the basic answer (select from the question database), profundity question resolved by the professor.

Exercise: Tutoring Agent selects the moderate exercise from the database based on the student instance.

Evaluate: Tutoring Agent gives directly. Or delivered to by the Professor Agent.

Student user entering process

1. User enter

Students and teachers enter in the system by inputting the user name/password.

Input: Output:

2. User selects the courses

After students entering, select the courses

After teachers entering, select the courses

3. Select phases

After students select the courses, select one phase among (preparation, discuss, question, exercise and evaluate)

4. Enter the preparation phase

Display the relevant courses section; introduce the emphasis and difficulty, the related bibliography

Choose them and check the details

Exit

5. Enter discussing section

Display the course's discussing section. If impossible electronic professor will recommend the discussing section

Enter the relevant discussing section

Discuss each other

Exit

6. Enter the question phase

Display the dialog box, write in the question

Affirm the referring

Display the answer or give out the information"please waiting for the answer from teachers"

Exit

7. Enter the practice phase

Select the local chapter practice or total practice

Select the index of practice difficulty

Submit the request

Send out corresponding topic

Do the exercise

Submit

Exit

8. Enter evaluating phase

Enter in.

Send out the examination.

Examination

Time end or early submit

Exit

6.3.2 User Agent——Teacher Agent

entry(identity validate)——courses select——functions select(discuss, put out the information, bulletin, homework, evaluation, depth answer)——implement—submit

Put out bulletin, homework, depth answer and the evaluation. Electronic professor collect the information and deliver to the Professor Agent.

Teacher user entering process

1. User entry

   Students and teachers enter the system through the user name/password.

Input: Output:

2. User selects the courses

   After student entry, select course

   After teacher entry, select course

3. Teacher select function phase

   After teachers select courses, then discuss, put out information (bulletin, homework, evaluation), depth answer.

4. Teacher enter discussing section

   Display different discussing section

   Choose and take part in the discuss

   Exit

5. Teacher enters and sent out information

   Display the bulletin and students' list

   Choose the new information and put out the textbox

   Fill in the new information in textbox (such as homework and bulletin)

   Choose the student list and put out the textbox, then fill in the suggestions, evaluation and expectation to the students

   Press ensure and submit information

6. Teacher enter the depth answer

After entry, display the question list

Choose and display the idiographic question

Give the idiographic answer in the textbox under the question

Press ensure and submit

CHAPTER 7

Conclusion

This thesis does the following work:

- Introduces the history and actuality of the ITS.
- Introduces the agent technology in the AI field.

In order to enhance the performance of ITS, the author proposed that Agent technology was applied to ITS.

- System modeling and database construct
- Analyses the function of the models in this system.
- Analyses the interaction relation between the models.

Finally the author concludes the use of agent in ITS as following. (chapter 2.4 include)

1 Individualization of distance teaching

2 Human-computer interaction and teaching method

3 Self-adaptive evolution of agent

4 Efficient use of teaching resources

5 Cooperation and intelligentization

6 Networking teaching helps cultivate learners' abilities and develop quality education

The future trend of ITS

ITS can not develop alone. It involves computer science, education, cognitive science, AI and many other subjects. At the present time, it shows the following trends:

1. The application of Intelligent Agent (IA)

An Intelligent Agent (IA) is an entity that perceives and acts according to an internal declarative body of knowledge. From the technical point of view, an Intelligent Agent is supported by many kinds of techniques and is quite practical. It is these features that are used by the developers to increase its performance and expand its functionality. In this way, an Intelligent Agent can automatically meet the requirements of different users. In ITS, learners can use IA to search effectively what they need. ITS has the learning function, which helps it search actively and effectively on the web and collect the relevant information. [20]It can thus solve many problems, such as how to use a single key word to search for information, how to get rid of the large amount of irrelevant information caused by the search engine, and how to achieve the precision of information search, etc... It does help improve the learner-instructor interaction and self-oriented study in teaching and learning process. In addition, this is a high efficient approach to search for information on the web.

2. Natural language processing (NLP) application technology

Natural language processing belongs to high-tech and is also the core topic of information processing. People have showed great interest in natural language processing technology. By adopting the theories and techniques of AI, computer experts manage to use computer program to reproduce the natural language. They are successful in forming a system which can understand natural language. Based on the system functionality, they set the standard to judge whether the computer can understand language, namely, input-process-output. In the research and development of ITS, especially in the intelligent human-computer interface, research findings on natural language processing can be used to enhance the intelligentization of the system. For instance, intelligent human-computer interface can improve the interaction between human and computers; semantic network can help express more clearly the hierarchical and semantic relationships between knowledge topics; fuzzy search can make full use of the knowledge base system; and machine translation can help cross language communication.

3. Virtual Reality Technology

Virtual Reality (VR) technology borrows from multimedia technology, simulation technology and computer science. It builds a virtual world with which people can interact. Virtual Reality technology can create a stronger illusion and it is an interactive experience. Teaching is a process of imparting knowledge, and personal experience can speed the learning process and help memorize the knowledge taught. In ITS, the user is placed in a virtual environment created, and can therefore exploit what they can not experience in reality. As a result, the teaching goal has been reached in this way. VR technology allows the interaction between learners and the existing information. Learners can experience in the artificial world various time and space, contact with all kinds of virtual objects and any part of the virtual world. This becomes a convenient way for learners to practice.

4. The application of Modern Learning Theory

Modern Learning Theory says that learning is not a passive process in which all information is to be recorded. Instead, it is an active process. It requires learners to play an active role in information processing. Traditionally, learners are quite passive in learning knowledge and are always treated with the cramming method of teaching. In the learning process, learners should learn to choose some information on their own and neglect those irrelevant. They should be able to understand the new information according to their own experience or different situations. The application of Modern Learning Theory in ITS can help provide sufficient information for learners in their study. It can trigger learners' interest in learning and self-oriented study. Furthermore, it can meet the individual requirements of different learners.

With the implementation of education reform, ITS will play an increasingly important role for certain. For the moment, there are still quite a few research findings in China. Few network teaching systems are used in teaching. So we should put more efforts in this promising field.

This thesis mainly focuses on the application of agent technology in ITS. More details are provided in the following part.

Agent technology is a hot topic in AI at present. People have been trying to use the agent technology to unify and further develop AI. Some even try to use it to unify and develop software. This shows that agent technology has a wide and promising application prospect. Thus in the application of ITS, we can utilize the multi-agent system. This is mainly because the independent and cooperative relations between agents feature largely the multi-agent system. And these features will help solve the complicated problems in ITS and support the realization of the complex functions of the teaching system.

BIBLIOGRAPHIES

[1] Joseph Psotka, Sharon A. Mutter (1988). Intelligent Tutoring Systems: Lessons Learned. Lawrence Erlbaum Associates. ISBN 0805801928.

[2] 引自 Foundations of intelligent tutoring systems, author: Martha Campbell Polson, J.Jeffrey Richardson, Elliot Soloway

[3] For an example of an ITS authoring tool, see Cognitive Tutoring Authoring Tools

[4] （Koedinger, K. R.; Corbett, A. (2006), "Cognitive Tutors: Technology bringing learning science to the classroom", in Sawyer, K., The Cambridge Handbook of the Learning Sciences, Cambridge University Press, pp. 61–78 ）

[5] http://en.wikipedia.org/wiki/Intelligent_tutoring_system

[6] Russell, Stuart J.; Norvig, Peter (2003), Artificial Intelligence: A Modern Approach (2nd ed.), Upper Saddle River, NJ: Prentice Hall, ISBN 0-13-790395-2, http://aima.cs.berkeley.edu/ , chpt. 2

[7] Huang-Min, Tong Zhen-Sheng. Research on the distributed multi-agent system. 2002

[8] Russell, Stuart J.; Norvig, Peter (2003), Artificial Intelligence: A Modern Approach (2nd ed.), Upper Saddle River, NJ: Prentice Hall, ISBN 0-13-790395-2, http://aima.cs.berkeley.edu/ , chpt. 2

[9] Fan Shun-Yu, Cao Jun-Wei. 《Multi-Agent System theory, technique and application》 Qing-Hua University publisher.

[10] N. Kasabov, Introduction: Hybrid intelligent adaptive systems. International Journal of Intelligent Systems, Vol.6, (1998) 453-454.

[11] http://en.wikipedia.org/wiki/Knowledge_representation

[12] http://en.wikipedia.org/wiki/Agent_communication_language

[13] He Yan-Xiang, Chen Zi-Meng. 《The design and application of agent and Multi-agent system》

[14] Fan Shun-Yu, Cao Jun-Wei. 《Multi-Agent System theory, technique and application》 Qing-Hua University publisher.

[15] Liu Hai-Long, Wu Hai-Jun. The distributed optimize arithmetic of Multi-Agent task assign based on contract net. Zhe-Jiang Universtiy transaction. 2001, 35th volume, 5th issue.

[16] Dong-Bin, He Bo-Xiong, Zhong Lian-Xiong. Research and development of the distributed artificial intelligence and multi-agent system. Xi-An engineering institute. 2000.

[17] Alex Rogers and E. David and J.Schiff and N.R. Jennings. The Effects of Proxy Bidding and Minimum Bid Increments within eBay Auctions, ACM Transactions on the Web, 2007

[18] Nathan Schurr and Janusz Marecki and Milind Tambe and Paul Scerri et al. The Future of Disaster Response: Humans Working with Multiagent Teams using DEFACTO, 2005.

[19] Ron Sun and Isaac Naveh. Simulating Organizational Decision-Making Using a

Cognitively Realistic Agent Model, Journal of Artificial Societies and Social Simulation.)

[20] Xu-Li, Han Xiao-Gang, Wang Huai-Min. Agent technology application in Internet. Computer engineering science. 1999(21)