IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Suite 200, Hershey PA 17033, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

ITP4565

A Methodology for Developing Complex Information System Based Multiple Intelligent Agents

Yao Li, Feng Xiaosheng, and Zhang Weiming
Department of Management Science and Engineering, School of Humanities and Management
National University of Defense Technology, Changsha, 410073, Hunan, China, PRC
Tel: 860(731)4225384
{Liyao, Xshfeng, Wmzhang}@nudt.edu.cn

ABSTRACT

This article presents a methodology for developing a complex information system by integrating Organization Theory with Multiply Intelligent Agents technology. The main idea of this method: the complex information system is regarded as a multi-agent organization, analyzed the organization characteristic of multi-agent systems according as computing organization theory, and built the organization model, accordingly formed the strict specification. Secondly all types of organizational roles taken on by multiple intelligent agents are fixed on according as techniques in existence and available resources, and the responsibility and granularity are also made sure from this. Then, conceptual models of all kinds of agents are designed by adopting Belief-Desire-Intention (BDI) architecture in according with the organization model, and a clear and operable development pattern (i.e. interaction model and BOS model) is built for realizing the system on computer. Finally, according to the conceptual models, the prototypal complex information system is got quickly by programming on a cooperative working platform (MBOS).

1. INTRODUCTION

Now agent technologies are being applied to the development of large-scale and complex commercial, industrial, military, educational and medical treatment information systems. Such systems are so complex, involving hundreds, perhaps thousands of agents, that we can design and implement them vary hard. Therefore there is a pressing need for Software Engineering techniques to support the analysis, design and implement processes of such complex information systems [1].

Software development methodologies have evolved from the class waterfall model, to a spiral model, to prototyping, to object-oriented and, recently, to scenario-based design [2]. However, all these existing methodologies are not suitable to develop an agent-based complex information system. Some techniques are difficult to adequately capture the autonomous problem-solving behavior, or the smart decision making of an agent, and some techniques fail to describe the richness of interactions between agents and the complexity of organizational structures in an agent-based complex information system, and so on. For these reasons, developing an agent-based complex information system requires a new conceptual framework and a new methodology for the analysis and design.

In this paper, we present a methodology for developing a complex information system by integrating Organization Theory with Multiply Intelligent Agents technology. We divide a development lifecycle of an agent-based complex information system into three phases: 1) Multi-Agent Organizational Design (MAOD) for building system organizational model at the macro level; 2) Agent-oriented Design (AD) for building agent conceptual models based on Belief-Desire-Intention (BDI) architecture[5], interaction model between agents, and BOS model at the micro level; 3) Multi-Agent Systems Realization (MASR) for implementing the multi-agent systems on a cooperative working platform (MBOS)[9]. A development lifecycle is shown in Figure 1.

In the phrase of MAOD, the complex information system is regarded as a multi-agent organization, analyzed the computing organizational characteristic of multi-agent systems according as organization theory, and built the organizational model, accordingly formed the strict specification. Then, all types of organizational roles taken on by multiple intelligent agents are fixed on according as techniques in existence and available resources, and the responsibility and granularity of each agent are also made sure from this.

In the phrase of AD, conceptual models of all types of agents are designed by adopting Belief-Desire-Intention (BDI) architecture in according with the organization model, and a clear and operable development pattern is built for realizing the system on computer.

In the phrase of MASR, the prototypal complex information system is programmed on a cooperative work platform, called MBOS, which includes agents building toolkit specifically tailored to the conceptual models of BDI agents.

The methodology we propose is appropriate for the large-scale and complex information system, especially for hiberarchy, with the following main characteristics:

(1) It is assumed that the goal obtained by a system that maximizes some global quality measure is consistent with the one of the system components.
(2) Agents are coarse-grained computational systems, each making use of significant computational resources.
(3) The overall system contains a comparatively large number of agents, perhaps hundreds of agents, which are distributed over the different geographic locations.

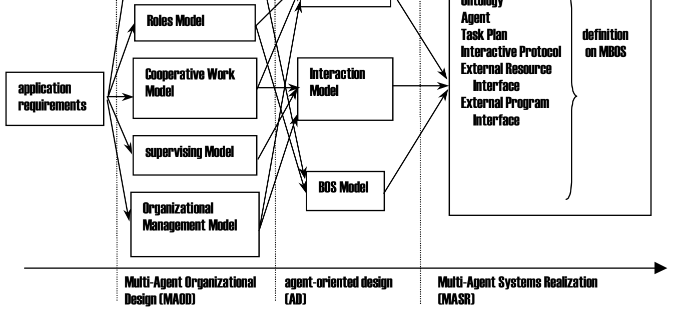


Figure 1. A development lifecycle of an agent-based complex information system.

The remainder of this paper is structured as follows. Multi-Agent Organizational Design is discussed in section 2, Agent-oriented Design in section 3 and Multi-Agent Systems Realization, related work and some conclusions are presented in section 4.

2. BUILDING AN ORGANIZATIONAL MODEL FOR MULTI-AGENT SYSTEMS

Our methodology is intended to allow a developer to analyze systematically requirements by building a multi-agent systems organizational model. The methodology borrows some terminology and notation from Organization Theory [4][5]. It provides an agent-specific set of concepts and models through which the developer view building an agent-based complex system as a process of organizational design. By this methodology, a software developer can understand and model an agent-based complex system precisely step by step.

In the methodology, a set of concepts of a multi-agent organization includes: organization, organizational benefit, organizational structure, organizational process, role, responsibility, authority, cooperation, control, and so on. Modeling a multi-agent organization can divide into modeling organizational structure and modeling organizational process, and the result of modeling includes five items: Organizational Chart, Roles Model, Cooperative Work Model, Supervising Model, and Organizational Management Model.

By this methodology, we will gain an artificial computational organization, which consists of multiply intelligent agents. And the intelligent agent is a computing entity, which is semi-autonomy and constrained by organizational principia [6][7].

2.1 Organizational Model

Organizational Model is composed of five constituents (see figure 1): Organizational Chart, Roles Model, Cooperative Work Model, Supervising Model, and Organizational Management Model.

Organizational Chart is used to render the main roles and the authority relationships in a computing organization. For example, Figure 2 is a part of organizational Chart of Multi-agent Intelligence processing Cooperatively Systems (MICS)[8]. In the vertical, it shows the authority and accountability relations, and in the lateral, it shows the divided work and cooperation relations.

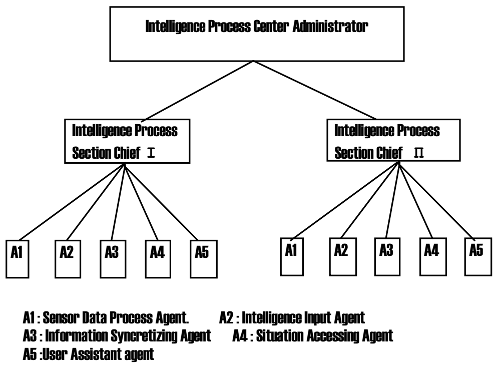

Intelligence Process Center Administrator
Intelligence Process Section Chief Ⅰ
Intelligence Process Section Chief Ⅱ
A1
A2
A3
A4
A5
A1 : Sensor Data Process Agent.
A2 : Intelligence Input Agent
A3 : Information Syncretizing Agent
A4 : Situation Accessing Agent
A5 :User Assistant agent

Figure 2. A Part of Organizational Chart

Roles Model specifies each role in an organizational Chart. Here a role is viewed as an abstract description of a computing entity's expected function. We can characterize each role by two types of attribute: responsibility and authority. Template for a role schema is as follows:

```
Role Name : <define the name of the role>
Role Description : <describe the role shortly>
Responsibility : <define the functionality of the role>
 Basic Skill Set : <define all the basic skills which the role must has
                              for undertaking its responsibility>
 Function Set : <define all the functions which the role must perform by
                              its basic skills or cooperating with other agents>
 Service Set : <define all the services which the role provides to the
                              public>
 Safety Condition Set : <list all the limit conditions about the role>
Authority : <define the rights associated with the role>
 Resource Permission : <define all the resources about knowledge or
                                     information that can legitimately be used to
                              carry out the role >
Monitor Permission : < define all the underlings and monitor contents >
```

Cooperative Work Model describes dependencies and relationships among the various roles in a multi-agent organization, which is central to the way in which the system functions. Cooperative Work Model defines the operational flow computationally, specifies all kinds of information needed for performing an operation, e.g. the flow initiator, the flow finish, each step in an operation process, information passing rules between steps, cooperative relations between roles, interactive protocols, etc.

Modeling cooperative work process is based on three kinds of analysis:

(1) Operation activity analysis: analyzing which activities the organization should has in order to achieve its goal. Only through analyzing operation activity strictly, you can make certain which tasks each role must perform, which tasks belongs to the same type, logic relations between the tasks, which task is important or imminent.
(2) Decision-making analysis: analyzing which kind of decision-making in each operation activity, which department or role to make the decision, which way to realize the decision-making, etc.
(3) Relation analysis: analyzing all the relationship about authority, responsibility, communication, and coordination in an operation flow.

Supervising Model builds a control mechanism according as the Cooperative Work Model. In Supervising Model, formal controls are designed to ensure that members of an organization act in ways that lead to the attainment of organizational goals. Control mechanism can be considered as either a part of organization structure or as a process. The basic elements of control mechanism are standard setting, measurement, and action. Control mechanisms consist of these three activities designed to ensure that tasks are carried out and completed in a desired manner. Standards specify desired states for all three steps in the task sequence. Control mechanisms then measure the actual state of operation and output. Through a comparison between standard and actual, the control process is designed to correct work that is not proceeding well by showing where adjustments need to be made.

In theory, we also can view the lifecycle of multi-agent organization as four stages consisting of initial birth, aggregation, regulation, and maturity. In our methodology, we attempt to divide the organizational change into two parts: large and complex change is made by system administrator; small adjust or adapt is made by intelligent agent which plays manager's role. By this ways, the organization is kept to be stability, persistency, and adaptability, from birth to maturity gradually.

Organizational Management Model is used to analyze and design organization process. It models the changes of activities going on within a organizational structure, by analyzing task uncertainty and resource, and constituting relevant organizational management strategies.

Five basic organizational management strategies, we think, are as follows:

(1) Responsibility Assignment strategies: defining all the roles distributing within agents.
(2) Agent Realizing strategies: defining each agent's granularity, skills, resources, relative location, main realizing techniques, etc.
(3) Optimizing Work Process strategies: defining optimal cooperative work strategies according to the changes of goal, task, environment, etc.
(4) Coordination Control strategies: defining some control strategies to improve system coherence and coordination
(5) Reliability strategies: defining some strategies to keep problem-solving reliable and secure.

2.2 The Analysis Process

We have designed a multi-agent organization modeling method [9][7]. A multi-agent organization can be modeled according to following steps.

1. Define the multi-agent organization goal and identify the goals of all administrative levels in this organization by a goal hierarchy diagram.

First, identify the collective goal, and then decompose it into goals of all administrative levels.

2. Identify all operation activities or tasks to realize goals, and classify them.

Developer should analyze all kinds of operation activities and available resources associated with them carefully. For example, all the tasks that Intelligence Process Section II in Figure 2 should accomplish are 1) Syncretizing multiply information sources; 2) Accessing situation cooperatively; 3) Providing information sharing service.

3. Design problem-solving and managing flow chart.

In our methodology, we have a diagram, called Problem-solving and Managing Flow Chart [9], to describe the problem solving activities recurrently flowing in a relatively stable program.

First, all operation activities are divided into long-term tasks and short-term tasks. For each long-term task, draw the problem-solving flow chart. Then, incorporate the same function in all problem-solving flow chart to form a department's problem-solving flow chart. Finally, for each short-term task, design associated work team, elaborate the full problem-solving flow chart.

Based the full problem-solving flow chart, design the control mechanism for the main task, gain the full elaborate Problem-solving and Managing Flow Chart.

4. Define managing roles and problem-solving roles.

According to the Problem-solving and Managing Flow Chart, study out each role's input object, output object, needed skills, functions, and services, and make out all the limit conditions about the role. At the same time, stipulate authority associated with the role's responsibility.

5. Build cooperative work model and supervising model.

Specify the details of Cooperative Work Model and Supervising Model according as the Problem-solving and Managing Flow Chart and Role model.

6. Set up full organization structure, and form Organizational Chart.

From now on, we have got a full organization structure, and we can draw the Organizational Chart.

7. Make certain agents taking on the roles.

We need to analyze the environment and techniques to realize an agent, and identify each agent taking on the roles. There is not necessarily a one-to-one mapping between roles and agents. If you identify each agent taking on the roles, you also make certain each agent's granularity.

8. Build organizational management model.

By analyzing task uncertainty and resource, and constituting relevant organizational management strategies, e.g. Responsibility Assignment strategies, Agent Realizing strategies, Optimizing Work Process strategies, Coordination Control strategies, Reliability strategies, and etc.

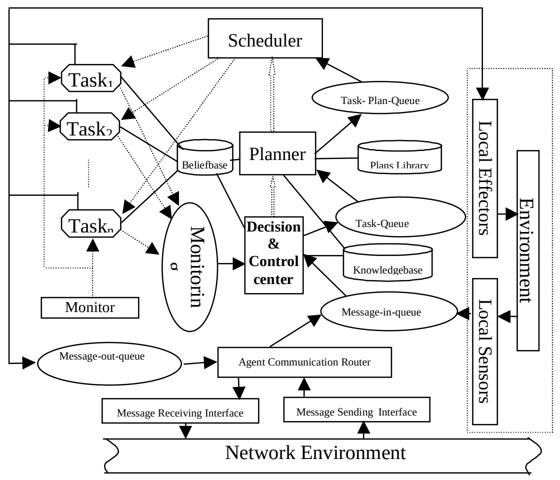
3. DESIGNING MULTI-AGENT SYSTEMS BASED ON BELIEF-DESIRE-INTENTION ARCHITECTURE

The aim of our design process is to transform the abstract organizational model into lower abstract models about what is required of each individual agent in order to take on its roles and how to interact with other agents in an organization. The design process involves three models for each type of agent. The *Agent Model* identifies one type of agent that takes on some roles, and is created from Roles Model, Cooperative Work Model, and Organizational Management Model. The *Interaction Model* specifies the details of interactions between agents, such as interactive protocols, interactive languages, and constrains in an interaction, which is created from Cooperative Work Model, Supervising Model, and Organizational Management Model. The *BOS model* documents a set of BOS (section 3.3), which a multi-agent organization consists of, and each agent must belong to one of a BOS.

3.1 Agent Model

In our methodology, we design and implement the agent with Belief-Desire-Intention structure (BDI), which is called CSA (Constrained-by-organization Semi-autonomic Agent, CSA), where the Semi-autonomic primarily refers to being capable of autonomous and flexible action under abiding by the organizational rules. The architecture of CSA is presented in Figure 3.

Each type of Agent Model is specified by Belief-Base, Goal-Set, and Plan-Library, specific to belief-desire-intention architecture. Belief-Base describes the information about the environment and internal state that an agent taking on some roles may hold. Goal-Set describes the goals that an agent may adopt, and the events to which it can respond. Plan-Library describes the plans that an agent may employ to achieve its goals or respond to events it perceives.



(In the figure, dashed stands for control flow; real line stands for information flow)
Figure 3. CSA Architecture.

Belief-Base consists of abstract entities and concrete entities. Abstract entities are self or acquaintance information, such as, basic skills, functions, services, authorities, acquaintance, and so on. Concrete entities are state information about a particular application domain. Some entities are defined from the Ontology of the application domain. All the entities are from the Roles Model, and these roles are that the agent is taking on. We use a Framework for representing the knowledge about each entity and its properties.

Goal-set is a set of goal and event formula signatures. Each formula in Goal-set is an operation on some elements in Basic skill set, or Function set, or Service Set, but Goal-Set must includes the operation on each element in Function Set. If not, the agent can't take the responsibility of the role. The agent performs a goal by finding the set of plans in plan-Library whose solving-goal matches the goal formula, and executing the plans to determine the success or failure. Apparently, the goals in Goal-set are compatible with the beliefs in belief-Base [14].

Plan-Library is a set of plan. Each element in Goal-Set at least matches the solving-goal of one plan in Goal-Set. That is to say, there are several plans that have the same solving-goal in Plan-Library. Each plan is specified by plan name, solving-goal, precondition, and a state transition diagram. A state transition diagram is interpreted as a recursive finite-state machine. A state transition diagram describes a plan how to achieve its solving goal. A state transition diagram has three types of node: start states, internal states, and end states, and one type of directed edge: behavior rules. Each behavior rule consists of two parts: conditions $\rightarrow$action. It means if the agent's belief satisfies the conditions then agent to take that action. All the actions belong to the agent's basic skills in its Belief-Base, or cooperative actions with the other agents, or executing a plan in Plan-Library.

3.2 Interaction Model

Interaction Model specifies interactive relationships between cooperation agents in MAS. In analysis stage, we have described coordination relationships and control relationships between roles by Cooperative Work Model, Supervising Model, and Organizational Management Model. Now we use these analysis results to design interaction details between agents taking on specific roles.

Interaction Model consists of interactive protocols, which is a structured pattern of interaction between agents. Each interactive protocol is defined by a finite automata and stipulations. The finite automata describe the specific executing steps to perform an interaction. The stipulations describe the interactive languages, ontology, and constrains about the interaction.

3.3 BOS Model

A social organization is generally composed of several smaller basic organizations. So, the cooperation exists both among these basic organizations and within each of them. For example, a university consists of many departments, and a department is divided into several teaching and researching sections or administrative sections. If a section is regarded as a basic organization, the organization is then composed of chief of the section, several staff members, and the public facilities.

In our methodology, we present a Basic Organization Structure (BOS) concept [12]. The basic organization cells, called BOS, are used to model the smallest groups of agents in a LAN. By this way, the interrelations among several agents and among groups of agents can be controlled more effectively; thus the system will run effectively and cooperatively as a whole and implement the corresponding global and local goals. BOS is mainly used to support the cooperative problem solving among the coarse-grained, loosely coupled, and groups of semiautonomous agents.

BOS model are simply used to model a multi-agent organization, and it is created from organizational Chart and Roles Model. For example, Figure 2 showing a part of organizational Chart of MICS can be modeled by three BOS as follows:

Intelligence-Process-Center=(Administrator; Section Chief , Section Chief)

Intelligence-Process-Section- = (Section Chief ; Sensor Data Process Agent, Intelligence Input Agent, Information Syncretizing Agent, Situation Accessing Agent, User Assistant agent).

Intelligence-Process-Section- = (Section Chief ; Sensor Data Process Agent, Intelligence Input Agent, Information Syncretizing Agent, Situation Accessing Agent, User Assistant agent).

4. CONCLUSIONS AND FURTHER WORK

Building large and complex information systems needs the support of integrative software developing environment. MBOS (Multiply Basic Organization Structure)[8] is a software platform for creating and deploying organizationally intelligent agents that can cooperate with other agents. MBOS means multiply Basic Organization Structure (BOS), which are the basic organization cells composing of complex organization. MBOS can be used to build the large and complex dynamic control systems as a multi-agent organization by BOS, which is combining fast and nested and consists of multiply agents. MBOS provides with some tools for ontology creation, agent creation, BOS creation, organization creation, Integrating preexisting software, etc., and it can support cooperating effectively between agents and between organizations. MBOS also have coordination control tools and visualization monitor tools, and can support the reengineering of organizational structure and the optimization of operation flow of complex information system. So that it makes a complex information system be developed gradually and incrementally. In our methodology, after the phrase of Agent-oriented Design, we have obtained all agent conceptual models, interaction models between agents, and BOS models at the micro level. So we can implement the multi-agent systems on the MBOS immediately.

Now, more and more research work is about agent-oriented software engineering. Much of them are associated with object-oriented analysis and design [1]. But we think the philosophy of agent-oriented approach clashes with the philosophy of agent-oriented approach, and agent-oriented software developing needs new methods. The methodology we presented is not the same as the Gaia [10] and MaSE [11] also. In the analysis stage of Gaia, the roles in the system are identified from individuals, departments within an organization, or organizations themselves. In the analysis phase of MaSE, the roles are defined from accomplishing some system level goals. In our methodology, you should design roles by Problem-solving and Managing Flow Chart. Our methodology is more suitable to develop the large and complex information systems with hierarchy organization in a complex and static environment.

The methodology we presented here is summarized from the development of MICS, and is especially for Multiply Intelligent Agent Developing Environment MBOS. So in the future, we should research this methodology thoroughly and practice it more and more.

ACKNOWLEDGMENTS

This research was partly supported by a project from NSFC, which Grant No. is 79800007.

REFERENCES

[1] M. Wooldridge and P. Ciancarini. Agent-Oriented Software Engineering: The State of the Art. In P. Ciancarini and M. Wooldridge, editors, *Agent-Oriented Software Engineering*. Springer-Verlag Lecture Notes in AI Volume 1957, January 2001.

[2] Tung Bui, Jintae Lee. An agent-based framework for building decision support systems. Decision Support Systems, 25(1999): 225-237

[3] Yao Li. MICS Technology Report. School of Humanities and Management, National University of Defense Technology. 2000.10

[4] Robey D. Designing Organizations. Richard D Irwin,INC,1986

[5] Yang Honglan, Zhang Xiaorong. Modern Organization Theory (in Chinese) . Shanghai: Fudan University Press. 1996

[6] Yao Li. Building the Organizational Model of DAI System. In Computer Engineering (in Chinese), 1997, 23(3), 15-19

[7] Yao Li, Zhang Weiming, Chen Wenwei, Wang Hao. Research on the Building Technology of Multi-Agent Systems. In Journal of Computer Research & Development (in Chinese), 1999(July), 36(Suppl): 50-53

[8] Yao Li, Zhang Weiming, et al. Multiply Intelligent Agent Developing Environment MBOS. In Computer World (in Chinese), 2001,7,23(28)

[9] Yao Li, Zhang Weiming. Intelligent and Cooperative Information Technology (in Chinese). Beijing: Publishing House of Electronics Industry, 2002: 168-181

[10] M. Wooldridge, N. R. Jennings, and D. Kinny. The Gaia Methodology for Agent-Oriented Analysis and Design. Journal of Autonomous Agents and Multi-Agent Systems. 2001, 3(1): 45-67

[11] Clint H. Sparkman, Scott A. Deloach and Athie L. Self. Automated Derivation of Complex Agent Architectures from Analysis Specifications. Proceedings of the Second International Workshop On Agent-Oriented Software Engineering (AOSE-2001), Montreal, Canada, May 29th 2001.

[12] Yao L., Zhang W., et al. (2002). Basic Organization Structure Model for Cooperative Information Processing. In: Mehdi Khosrow-Pour. Eds. Issues and Trends of IT management in Contemporary Organizations. Idea Group Publishing. 836-839.

[13] Wooldridge M., Jennings N.R.& Kinny D. (2000). The Gaia Methodology for Agent-oriented Analysis and Design. Journal of Autonomous Agents and Multi-Agent Systems. 3(3), 285-312.

[14] Michael Wooldridge. Intelligence Agent. In Gerhard Weiss, eds. Multiagent System: A Modern Approach to Distributed Artificial Intelligence. the MIT Press,1999,27-78

[15] M. Wooldridge, N. R. Jennings, and D. Kinny. The Gaia Methodology for Agent-Oriented Analysis and Design. In Journal of Autonomous Agents and Multi-Agent Systems. 3(3):285-312. 2000.