MODEL ORGANIZATION CONSTRAINTS IN MULTI-AGENT SYSTEM

Xinjun Mao and Jiajia Chen
Department of Computer Science, National University of Defense Technology, China. Email:xjmao21@21cn.com

Abstract: The organization concept is an important abstraction to analyze and design multi-agent system. In this paper, we argue, the organization constraints should be explicitly modeled and reasoned when developing multi-agent system. The characteristics of organization constraint are discussed, a systematic method to model and reason the organization constraints is put forward and a case is studied.

Key words: Multi-agent system, Organization Constraint, Maintenance Goal

1. INTRODUCTION

Agent-oriented computing is rapidly emerging as a powerful paradigm for developing complex software system [1]. It provides us a number of high level abstract concepts to understand, model, analyze and design the entities and interactions in the complex system such as autonomous agent, social coordination, etc. Based on these concepts, there have been several attempts to develop the methodologies in support of analysis and design of multi-agent system such as Gaia[4], MaSE[5,6], AUML, etc. Recently, more and more agent-oriented methodologies borrow concepts and ideas from sociology and organization discipline to model and analyze multi-agent system such as role, responsibility, permission, organization, etc [1,2,3,4,6], based on which multi-agent system is modeled as an organization where there are a number of roles, each role has its tasks, responsibilities, and goals, different roles need to interact with each other to achieve their design objectives.

The organization abstraction and metaphor is important when developing multi-agent system. First, more and more multi-agent systems nowadays are intended to support real-world organizations, such as electric commerce, enterprise workflow management, etc. In such cases, an organization-based abstract and model can reduce the conceptual distance between real-word application and the software system. Therefore, it simplifies the development of the system and supports modeling the system in a natural fashion. Secondly, the organization abstract gives us a way to incorporate such high level concepts as role, responsibility, rule, etc to understand and model multi-agent system, which enables us to specify and analyze multi-agent system without considering the low-level and heterogeneous information. Such a way is natural and easy to understand not only for the software developer, but also for the stakeholders, users, etc.

In real-world, organization constraint is one of the organization elements and widely exists in applications in order to restrict the behaviors of autonomous agents in the organization. It defines the operation rules and guarantees the organization to behave in a consistent and coherent way. To model such organization constrain explicitly is of particular importance to understand, specify and analyze the system requirements, and further guide the design and implementation of multi-agent system.

Although organization metaphor and abstraction has gained many attentions, and organization rule concept has been integrated into some agent oriented methodologies like Gaia, MaSE, etc., there is little work to systematically analyze the characteristics of organization constraints. The organization constrains in multi-agent system are maybe complex, therefore it is necessary and helpful to provide a systematical method to model, analyze and reason them. The reminders of the paper are structured as follows. Section2 explains the characteristics of organization constrains. A systematical method to model and reason organization constraints is put forward by extending *i** framework in section3. Section4 introduces the related works. At last conclusions are made.

2. CHARACTERISTICS OF ORGANIZATION CONSTRAINTS

In order to discuss the organization constraint and its characteristics, let us firstly consider a simple example from electric commerce application. Some company intends to develop a web system (called as InfoCenter system) based on Internet to publish the consuming and producing information about some kinds of products (e.g., computer sale, etc.) for its customers all around the world. The users that are the customers of the company and should have a valid account can post its consuming or producing information in the system. System can also automatically and actively send the posted information to any valid users who may concern about by semantics-based match analysis. Therefore, the consuming and producing information can be obtained and shared by the relevant customers in the system and they can further to conduct some business deals such as negotiations or bargains. In addition, the application has the following global constraints information: (1) All users who want to post information should have a unique and valid account; (2) The user should be truthful, he can not post the incorrect and unknown information; (3) system should guarantee that the valid user should log on the system before posting information. In general, organization constrain has such characteristics described as follows.

– Restricting system's elements

In most of cases, the organization constraint will restrict the elements in the system. Particularly, it will restrict the behaviors and plan choices of agents and the interactions among them in multi-agent system. For example, according to the third constrain in the sample, system can not permit the user to post information if he has not logged on.

– Global

In general, the organization constrain is global and will restrict all of the elements in the organization.

– Mutually Consistent

The multiple organization constrains in one organization should be consistent. It will be impossible that an application has organization constraints φ and $\neg\varphi$. If that so, the agents in the organization will be confused and don't know well which organization constraints should be obeyed and the system may disorder. For example, you should not require that the users be not only truthful but also untruthful. However, the organization constraints of multiple organizations may be inconsistent.

– Satisfiable

The organization constraints should be satisfiable, which means the agents in the organization should have some effective way to obey or respect them. The unsatisfiable organization constraints will be meaningless because they will never be obeyed.

– Consistent With Agent's Goal

The organization constraints should be consistent with the agent's goal. If the inconsistency between organization constraints and agent's goals takes place, agents will be confused and be unable to make decision about their behaviors. Therefore, there should have some ways to solve the inconsistency, which may be agent-dependent and application-dependent.

For example, if the organization constraints have high level priority, then agent should modify its goals to adapt the organization constraints in order to solve the inconsistency between them.

- Non-conflict With Agent's Goal

The organization constraints should be non-conflict with the agent's goal, which means that the abidance of organization constraints should not hinder the agent's goals from being achieved, and vice versa. If the conflict between organization constraints and agent's goal takes place, there should have some ways to resolve the conflict, which may also be application-dependent and agent-dependent.

- Mutually Non-Conflict

The multiple organization constraints should not be conflict. The conflict between multiple organization constraints means that the abidance of one organization constraint will definitely hinder the other from being obeyed. If that so, it signifies that there are some problems in the system requirements about organization constraints.

- Stable

The organization constraint embodies the organization setting and generally remains constant throughout the analysis and design process. This is in contrast to other elements such as system structure, agent's goals, plan choice, belief, etc., that may be dynamically changed from time to time.

- Persistent

Generally, the organization constraints will persist during the life cycle of system. This is in contrast with the other system elements such as goals, tasks, etc that will be changed during the life cycle of agent and system.

3. MODELING ORGANIZATION CONSTRAINTS

In this section, we will introduce the method to model the organization constraints in a structural and systematic fashion by extending *i** framework. The *i** framework was once developed for modeling and analyzing organizations to help support business process reengineering and requirements engineering. The framework focuses on modeling intentional and strategic relationships among actors in the organizations. It consists of two types of models: the Strategic Dependency (SD) model and the Strategic Rationale (SR) model. The SD model is used to capture and specify the actors in the system and the network of relationships that hold among them. The SR model describes in more detail the alternative methods that the actors have for accomplishing their goals and tasks, etc [7,8].

A SD model is a graph consisting of nodes and links among the nodes. Each node in the model represents an *actor*, and each link between the actors represents the *dependency* among them, which specified how one actor depends on another for something in order to accomplish its goals and tasks. An actor in the model is an active entity that carries out actions to accomplish its goals and can be further differentiated into the concepts of *role*, *position*, and *agent*. A *role* is an abstract collection of coherent abilities and expectations. A *position* is a collection of roles that are typically occupied by one agent. An *agent* is an actor that has concrete manifestations such as human, hardware, or software, or combinations thereof. Four types of *dependencies* (i.e., *goal-, task-, resource- and softgoal-dependency*) are distinguished for indicating the nature of the freedom and control in the relationship among actors. A *softgoal* is similar to a *goal* except that the criteria of success are not sharply defined. The SD model provides one level of the abstraction for describing organizational environments and their embedded information system. It shows the macro or external (but nevertheless intentional) relationships among actors, while hiding the intentional constructs with each actor. It is useful in helping understand organizational and systems configurations as they exist, or as proposed new configuration.

The SR model provides a more detailed level of modeling by looking "inside" actors to model internal intentional relationships. It shows the micro or internal constructs and relationships in actor. The SR model is a graph consisting of four main types of nodes: *goal*, *resource*, *softgoal* and *task*, which appear in the SR model not only as external dependencies, but also as internal elements linked by some relationships. There are three main types of links in SR model: *means-ends*, *task decomposition* and *contribution* link. *Task decomposition* link describes how a task can be decomposed into s number of subtasks. *Means-ends* link specifies how a goal may be accomplished. *Contribution* link describes how one node contributes to another node. The SR model provides a way to model stakeholders' interests, and how they might be met, and the stakeholders evaluation of various alternatives with respect to their interests.

3.1 Modeling Organization Constraints in Strategic Dependency Model

The first question related with organization constraint when analyzing and specifying the multi-agent system may be that *what* the organization constraints are in the system. Such a question is important for the stakeholders and the requirement analyzers to understand the organization constraints clearly and can be served as the basic and precondition to elaborate on and refine the organization constraints. The question should be dealt with in the first step to model the organization constraints.

In the macro level, the organization constraints should be specified when constructing the strategic dependency models of applications. In this step, the requirement analyzers should consider not only what the actors (e.g., stakeholders and system itself) that the application has and the dependencies among them, but also whether there exist some organization constraints and what they are. The whole applications can be regarded as an organization, and the organization constraints of it should be specified explicitly and the relationship between the organization constraints and the dependencies of stakeholders should also be considered.

In order to support modeling the organization constraint in the macro level, the strategic dependency model in *i** framework is extended. A modeling element to denoting organization, which has explicit constraint specification and border to distinguish the actors in the organization that will be affected by the organization constrains from the actors outside that will not, is introduced (see *Figure1*).

In addition, the requirement analyzer should investigate whether the organization constraints specified are necessary and natural, and there are any inconsistencies and conflicts between them. There may need some tradeoffs about the degree of the organization constraints. Too strong constraints may lose the autonomy and proactive properties of agents in the organization and further lose the flexibility of the system. Too weak constrains, however, will impose little influences on the autonomous agent's behaviors. The macro modeling of the organization constraints are important, as well-defined specification and tradeoff of organization constraints will facilitate the micro-level reason and specification of organization constraints.

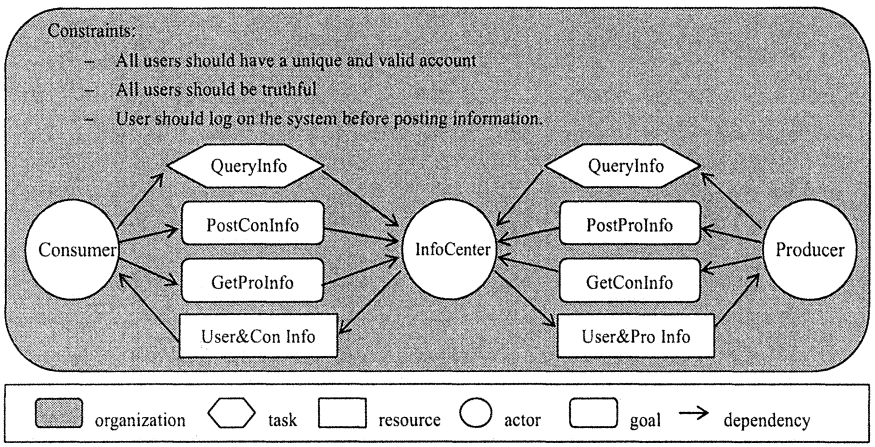

Figure1. The Extended Strategic Dependency Model

For the example described in section 2, we can identify the following three actors in the early requirement analysis phase: Consumer, Producer and

InfoSystem, where Consumer and Producer actors are actually the stakeholders of the system. They together constitute an organization and will be influenced by the organization constraints. The Consumer depends on InfoCenter to achieve goals of PostConInfo (denoting posting the consuming information) and GetProInfo (denoting getting the producing information), and to fulfill the task of QueryInfo (denoting querying the information). However, InfoCenter depends on Consumer to get the resource such as the Consumer user's information and the consuming information to be post. The organization constraints also are depicted in figure 1 as a part of organization specification.

3.2 Modeling Organization Constraints in Strategic Rationale Model

After specifying what the organization constraints are, the second question that should be dealt with may be that *how* the organization constraints influence the actors. Such a question is important to elaborate on and refine the organization constraints, helpful to investigate the actors' behaviors and valuable to guide the system design. Therefore the question should be dealt with as the second step to model the organization constraints based on the first step result

In the micro level, the organization constraints should be further refined, reasoned and specified when constructing the strategic rationale model of system, which provides detail information to model the inside intentional information about actors. In this step, the requirement analyzers should consider how the organization constraints will restrict the behaviors of the actors in the organization. Therefore, they should not only investigate the goal, resource, softgoal and task that each actors have, and the relationships among them such as *means-ends* and *task decomposition*, but also reason and consider how the organization will influence the intentional constructs and relationships of them in the actors.

In order to support modeling the organization constraint in the micro level, the strategic rationale model is extended. Two different types of goal will be differentiated: achievement goal and maintenance goal. The achievement goal means to achieve some state of affairs, and actually corresponds to the concept of goal in strategic rationale model. The maintenance goal means to maintain some state of affairs, and is a new modeling element to be introduced into the extended strategic rationale model. The maintenance goal is the abstract representation of organization constraints in the actor, therefore acts as one part of actor's internal intentional structures (see *Figure 2*).

The mapping from the macro level model of the organization constraints to the micro level model of the organization constraints is very simple. Each item of the constraint organization in the strategic dependency model will be mapped as the corresponding maintenance goal of the related actors. However, more work should be performed in this step to specify and reason the organization constraints. The relationship between the maintenance goal and the other elements of actor such as achievement goal, tasks, etc., should also be explored and specified. For example, some maintenance goal may influence and contribute positively to some achievement goal of actor. Moreover, the requirement analyzers should investigate whether there are inconsistency and conflict between the maintenance goal and the task, achievement goal, etc in the actor. If that so, the negotiation will be needed to resolve the inconsistent and conflict requirements.

Figure2 shows the strategic rationale model of the sample focusing on the InfoCenter actor. The organization constraints depicted in strategic dependency model will be mapped as the maintenance goal of the related actors. For example, the Consumer actor has the maintenance goals such as being truthful and logged before posting; the InfoCenter actor has such maintenance goals as logged before posting for users and having unique account for users. The maintenance goal of having unique account for users will contribute positively to the maintenance goal of logged before posting for users, which will further restrict the achievement goal of "post information".

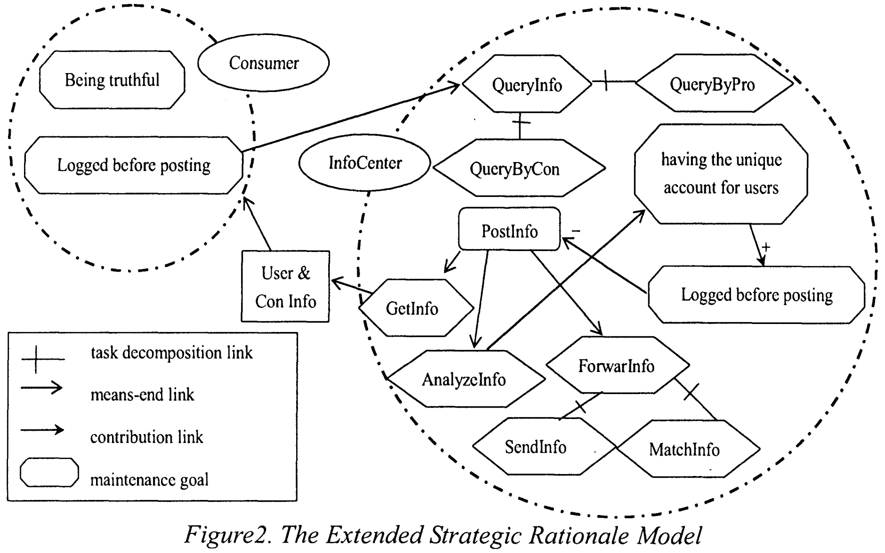

Figure2. The Extended Strategic Rationale Model

4. RELATED WORKS

Shoham in [1] discussed the concept of social law for artificial agents in computational environment, which will guarantee the successful coexistence of multi-programs and programmers. He presented ways to the off-line design of social laws, and pointed out the social laws will on the one hand constraint the plan available to the programmer, but on the other hand will guarantee certain behaviors on the part of other agents. Zambonelli believed that the organization abstraction should play a central role in the analysis and design of multi-agent system, and introduced three additional organizational concepts: organizational rule, organization structure and organizational pattern that are necessary for the complete specification of computational organizations [2, 3]. He also discussed the importance and necessity of explicit identification of organization rules in the context of open agent system. In [4], Zambonelli further extended the Gaia methodology by introducing the modeling activity to identify, specify and analyze the organization rule in the requirement analysis phase. The organization rules, he pointed out, express the constraints on the execution activities of roles and protocols and are of primary importance in promoting efficiency in design and in identifying how the developing MAS can support openness and self-interested behavior. Another important work recently about the organization rule is [6]. DeLoach further extended the MaSE, an agent oriented methodology introduced in [5], by borrowing the concept of organization abstraction and integrating organizational rules into existing multi-agent methodology. Especially, he investigated how to integrate the existing abstractions of goals, roles, tasks, agents, and conversations with organizational rules and tasks. He used the approach similar to the notions used in the KAOS to represent the organization rules. It is not the first time for us to present the concept of maintenance goal. In [14], we had presented the concept of maintenance intention to investigate the agent's behaviors. The concept of maintenance goal can also be found in the formal Tropos [9] and KAOS.

5. CONCLUSION

The organization concept is an important abstraction to analyze and design multi-agent system. Organization constraint is widespread in multi-agent system and will restrict the behaviors of agents in multi-agent system. The organization constraints have a number of important characteristics such as stable, consistent, non-conflict, persistent, etc. For complex multi-agent system, the specification and analysis of organization constraint may be difficult. In this paper, a structural and systematic way to model and analyze the organization constraints is presented by extending the *i** framework, an approach to model and reason the early requirement of system. In the macro level, the requirement analyzer should specify what the organization constraints in the system. Therefore, an organization modeling element, with explicit constraints specification and organization border is introduced into the strategic dependency model. In the micro level, the requirement analyzer should specify and reason how the organization constraint will restrict the behavior of agents in the multi-agent system. Therefore, an abstract cognitive concept of maintenance goal is introduced into the strategic rationale model. The steps are also described and a case is studied to show how to specify and analyze the organizational constraints from the macro level to the micro level.

6. REFERENCES

1. Y Shoham and M tennenholtz, On Social Laws for Artificial Agent Societies: Off-Line Design, Artificial Intelligence, Artificial Intelligence 73(1-2), 1995.
2. F. Zambonelli, N. Jennings, M. Wooldridge, Organizational Rules as an Abstraction for the Analysis and Design of Multi-agent Systems , Journal of Knowledge and Software Engineering, 11(3), 2001.
3. F Zambonelli, N R.Jennings, M Wooldridge, Organizational Abstractions for the Analysis and Design of Multi-agent System, AOSE'2001, LNCS2222, Springer, 2002.
4. F. Zambonelli, N. R. Jennings, and M. Wooldridge, Developing Multiagent Systems: The Gaia Methodology, ACM Transactions on Software Engineering Methodology, 12(3), 2003.
5. S A. Deloach, M F.Wood, and C H.Sparkman, Multiagents Systems Engineering, International Journal of Software Engineering and Knowledge Engineering, 11(3), 2001.
6. S A. DeLoach, Modeling Organizational Rules in the Multiagent Systems Engineering Methodology, Proc. of the 15th Canadian Conference on Artificial Intelligence, Alberta, Canada. May 27-29, 2002.
7. E. Yu, Agent-Oriented Modelling: Software Versus the World, Proc. Of Agent-Oriented Software Engineering, LNCS 2222, Springer-Verlag, 2001.
8. E Yu, Towards Modeling and Reasoning Support for Early-Phase Requirements Engineering, Proceedings of the 3rd IEEE Int. Symp. on Requirements Engineering, 1997.
9. A. Fuxman, et.al. Specifying and Analyzing Early Requirements: Some Experimental Results, Proceedings of the 11th IEEE International Requirements Engineering Conference, 2003.
10. Wooldridge, N R.Jennings, and D.Kinny, The Gaia Methodology for Agent-Oriented Analysis and Design, International Journal of Autonomous Agents and Multi-agent System,3, 2000.