



UNIVERSITY OF  
CALGARY

## Role model analysis method in Agent based system

**Prepared by: Li Jiang**  
jiangl@enl.ucalgary.ca

**Course Instructor: Dr. Behrouz H. Far**

**Abstract:** This paper provides a tutorial for the role model analysis in Agent based system development. It combines current available research results in the role modelling for the Agent based system and provides general method for the role model analysis.

**Key words:** Agent, agent based software engineering, role, role modelling, OO,

### 1. Introduction

Just as other methodology, Agent-Oriented Software Engineering is one of the most important contributions to the field of Software Engineering. It has some benefits compared to existing development approaches. Particularly, the ability to let agents represent high-level abstractions of active entities in a software system makes it easier to understand [12].

There are a number of Agent-based methodologies available rights now, such as Gaia methodology [1, 2], Multiagent Systems Engineering Methodology [3, 4] and etc. Each of these methodologies provides some ways to describe the agent system analysis and design. Among them, role-based analysis and design is one of very effective methodology for the agent system analysis and design. This document will focus on the introduction about the role-based analysis for the agent system design.

## 2. What is the role and Role Modelling?

In every organization in the society, people play various roles to implement various functions of the system; the roles might be “president”, financial manager or secretary and etc. Each role in the organization has certain responsibilities and rights; besides, each role also has certain way to communicate to each other in order to implement their responsibilities. After detail analysis of the characteristics of each role, following commonalities have been identified:

- ❑ Each role has certain responsibilities
- ❑ Each role has certain rights in the organization in order to function properly.
- ❑ Each role has certain way to communicate with other roles in order to make the system work properly
- ❑ Each role has its own initiative and goal to finish certain assigned work and do creative work intelligently.

According to Y. Shoham, an agent is a software entity that functions continuously and autonomously in a particular environment [7]. With comparison to this definition, we will find some analog to the concept of the role defined in the organization, the idea of “role” based analysis method in the agent system development is the same, and we will find that role model is a very nature way to describe the agent system function and its behaviours. And it is also very easy to understand the conceptual differences among agents based methodology with that of other methodologies such as Object-oriented methodology.

In object-oriented system, a role defines a position and a set of responsibilities within a role model [5]. The responsibilities of a role are consisted of services and tasks. A role has collaborators with which it interacts. This role definition reveals the static behaviour of the object in its nature,

In agent based system, a role model identifies and describes a prototype or recurring structure of interacting entities in terms of roles, and a role defines a position and a set of responsibilities within a role model [9]. So far, a widely accepted definition for role is: “A role is a set of capabilities and an expected behaviour” [6]. This definition reveals following points which related to the agent concept and its nature:

- ❑ The capability represents a set of actions that needed for the agent to achieve its task.
- ❑ The expected behaviour represents a set of events that an agent is expected to manage in order to “behave” as requested by the role it plays. Interactions among agents are represented by a couple action-events.
- ❑ The expected behaviour could be at any intelligent level.

Based on the discussion above, we can say that concept of Agent and the concept of Role is quite similar and complementary with each other. So role modelling for the agent system is nature way for the agent system analysis and design.

### 3. Why role modelling?

The advantages of the Role modeling over other methodologies are that agent offers a promising approach for agent analysis and design; this is because that [8]:

- (1). Role modeling reflects the social nature of the Agent which emphasize on social or interactive behaviour.
- (2). Role modeling reveals proactive property of agent since roles in a role model works together to accomplish a goal.
- (3). Role modeling provides a unified model through which Agents, objects, and people can play roles.
- (4). Role modelling encourage partitioning through which complex agent behaviour can be partitioned into roles.
- (5). Role modelling emphasizes design and role model synergy or synthesis would be valuable for agent design.
- (6). Role modelling reveals the dynamics nature of the role. Agent can take on various role models, such as organizations, formations. In role modeling, role assignments could be in dynamic way.
- (7). Role modeling provides documentation mechanism from which documentation for agent frameworks can be kept and this documentation can then be served as patterns for the agent analysis and design later on.

### 4. Agent system Analysis with role modelling

It is well-known that the objective of the analysis stage in software engineering is to develop an understanding of the system and its structure based on the requirements obtained. In the role modelling approach, this understanding is captured in the roles, its relationship and its interaction among each other. We view an organisation as a collection of roles that stand in certain relationships to one another, and that take part in systematic, institutionalised patterns of interactions with other roles. To define an organisation, it therefore suffices to define the roles in the organisation, how these roles relate to one another, and how a role can interact with other roles [11].

Normally, in role modelling, the analysis process for agent system will have following steps:

#### **Step 1: Identify various roles of the system from different dimensional perspective:**

In role modelling, an agent normally have been represented from many dimensional perspectives as shown in Table 1, it reflects how agents encompass behavior from objects, organizational theory, and artificial intelligence. One of the main tasks in this step is to identify as more roles in the system as possible from different perspective as shown in Table 1.

| <b>Dimensions of role model</b>     | <b>Context</b>   |
|-------------------------------------|--|
| <b>Responsibilities</b>             | services, tasks, goals, obligations, interdictions                                   |
| <b>Collaborators</b>                | roles it interacts with  |
| <b>external interfaces</b>          | access to services   |
| <b>relationships to other roles</b> | aggregation, specialization, generalization, role sequences                          |
| <b>Expertise</b>                    | ontology, inferencing, problem solving Knowledge                                     |
| <b>coordination and negotiation</b> | protocol, conflict resolution, knowledge of why other roles are related, permissions |
| <b>Other</b>                        | resources, learning/ adaptability  |

**Table 1: Dimensions of an Agent Role**

**Step 2: Analysis the relationship and interactions among various roles in the system.**

In this step, we must analyze various relationship, its interaction, its constraints and its rights among various roles in the system based on the content listed in following Table 2.

| <b>Attributes for the analysis</b> | <b>context</b>  |
|------------------------------------|---|
| Responsibilities                   | services, tasks, goals, obligations   |
| Interactions                       | How many interactions the role has with other roles it interacts with   |
| Rights                             | What right it has to performs the role  |
| Constraints                        | Its constraints conditions should be identified in the system, such as what it can not do, what action it should be restricted and etc.   |
| coordination and negotiation       | <p>What protocol the role used to interact with each other and how to map this protocol in the future system. Conflict resolution, knowledge of why other roles are related and etc.</p> <p>The protocol normally includes following contents [11]:</p> <ul style="list-style-type: none"> <li>• purpose: brief description of the nature of the interaction (e.g. “information request”, “schedule activity” and “assign task”);</li> <li>• initiator: the role(s) responsible for starting the interaction;</li> <li>• responder: the role(s) with which the initiator interacts;</li> <li>• inputs: information used by the role initiator while enacting the protocol;</li> <li>• outputs: information supplied by/to the protocol responder during the course of the interaction;</li> <li>• processing: brief description of any processing the protocol initiator performs during the course of the interaction</li> </ul> |

**Table 2 The analysis guidance for role attributes**

In the analysis process, we either can analyses the role model from scratch or we can use some kinds of pattern existed so far. For example, in Figure 1, a Supply Chain (SC) pattern can be used to analysis the new system.

A supply chain is a common pattern of collaboration for agents. Normally, it assumes that a supply chain has suppliers and consumers. A consumer can have many suppliers, but a supplier usually only has one consumer in any given supply chain. At the highest level, a supply chain is made up of SC Predecessors and SC Successors. A predecessor can have many successors. As shown in the role model in Figure 1, an SC Participant is both a predecessor and a successor, while an SC Head is a specialization of a predecessor, and an SC Tail refines a successor.

Figure 2 depicts a role sequence, and nested or contained roles are shown. In Figure 2, an SC Predecessor goes from being a Customer who negotiates for services to a User who just purchases and utilizes them. An SC Successor is first a Provider and then a Supplier. During negotiation, the Customer in the predecessor negotiates with the Provider in the successor. During delivery, the Operator in the successor delivers supplies to the User in the predecessor.

During the analysis process, various notations could be used for aiding the analysis. In the following examples in Figure 1 and 2, we assume that:

- A rounded box represents a role;
- An arrow represents the collaboration between roles and the direction of message it transmits.
- A triangle means that a role is refined,
- A filled circle signifies that a role can be played by more than one entity at the same time.
- A filled chevron signifies the order of the sequence

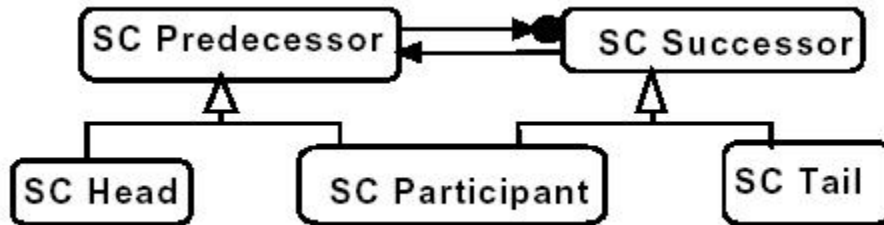


Figure 1: Supply Chain: Top Level Role Model

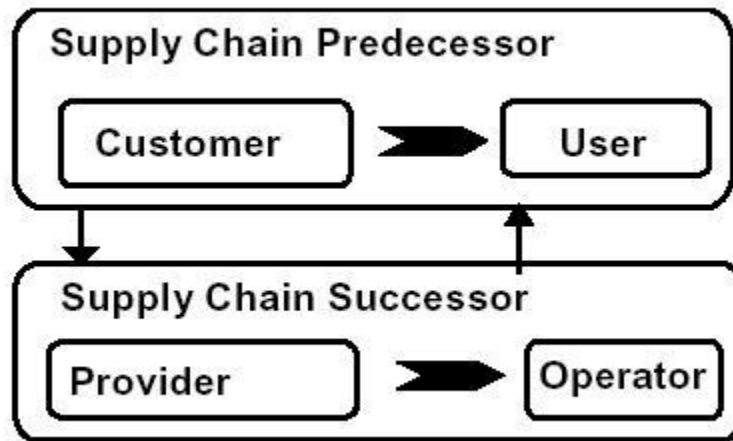


Figure 2: Supply Chain: Role Sequences

Another role model of relevance to agent systems is the Bureaucracy role model pattern. This pattern is found very useful in software systems. It captures the structure of human bureaucracies. In a bureaucracy, there is a long chain of responsibility, a multilevel hierarchical organization, and centralized control [10]

The Bureaucracy role model shown in Figure 3 has five roles: Director, Manager, Subordinate, Clerk, and Client. A Director manages the entire organization. Managers report to the Director, supervising the activities of their Subordinates. Due to the multiple levels in the Bureaucracy, intermediate level Managers have other, lower level, Managers reporting to them as Subordinates. The lowest level role in the Bureaucracy is a Clerk; these entities perform the actual work or service for a Client.

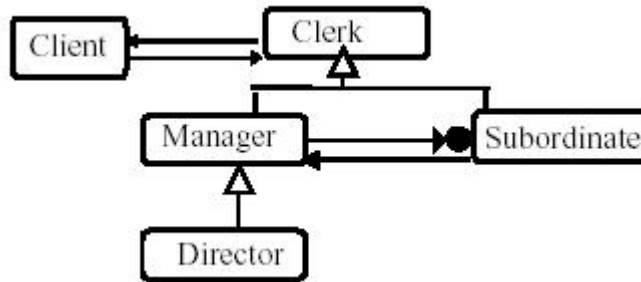


Figure 3: Role Diagram for the Bureaucracy Role Model

Actually, in my opinion, notations in UML can be used to better describe various relationships during the role modelling. A UML fashion representation of various role relationships is illustrated in Figure 4

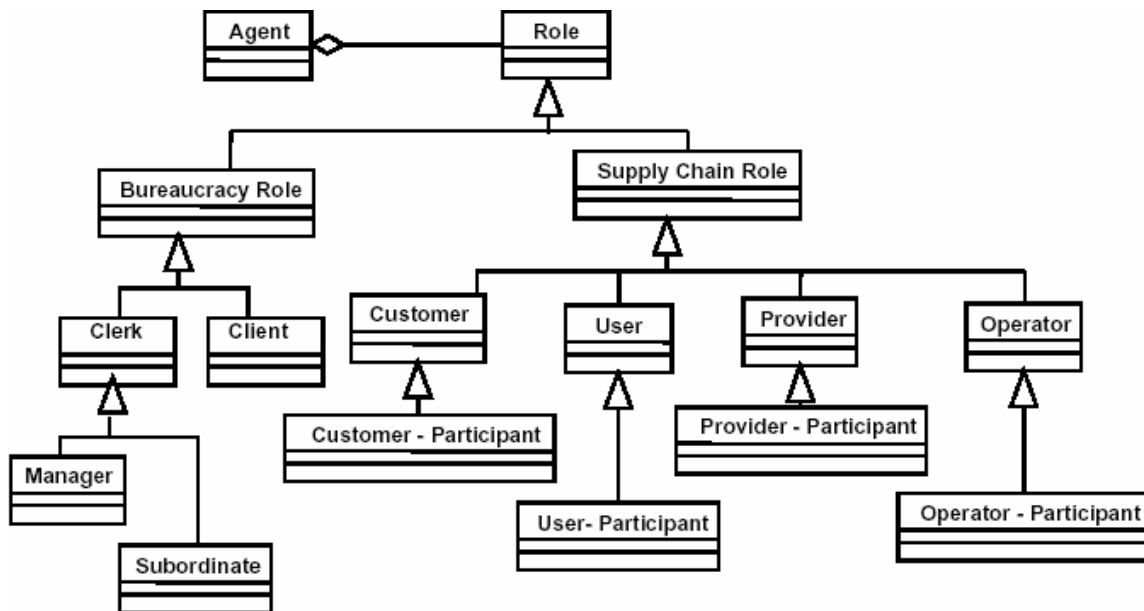


Figure 4 Role model represented in UML notation for Supply Chain & Bureaucracy Agent

**Step 3: Documentation the analysis results**

After analysis the roles and its relationship among each other, a good documentation for various roles, its relationships and interaction is needed. During the documentation, various kinds of forms or tables based on Table 1 and Table 2 can be used to capture the roles, its right, constraints, relationships and interactions. Any combination and extension of the Table 1 and Table 2 is encouraged.

For example, we can use following format to documents roles such as the format in Table 3:

|                           |  |
|---------------------------|--|
| <b>Role name:</b>         | AccountCenter Agent  |
| <b>Brief Description:</b> | The customer uses this agent to get into ER system.  |
| <b>Precondition(s):</b>   | The customer's information has been stored in the database   |
| <b>Post condition(s):</b> | Customers go into next corresponding agent   |
| <b>Process Steps</b>      |  |
| <b>1.</b>                 | The old user put into his/her login name and password through the interface  |
| <b>2</b>                  | Accountcenter Agent goes to DB Agent to check the information, and then validate this user.  |
| <b>3</b>                  | Accountcenter Agent leads the valid user into next corresponding agent   |
| <b>4</b>                  | Accountcenter Agent creates new account for the new user and save new user's information into database through DB Agent            |
| <b>5</b>                  | Accountcenter Agent daily checks database through DB Agent to remove inactive account if an account is inactive more than 2 years. |
| <b>6</b>                  | Accountcenter Agent daily checks database through DB Agent to check the expired Ad to notify the user who post this AD.            |
| <b>Exceptions</b>         |  |
|                           | Accountcenter Agent <b>will reject invalid user to go into the system</b>  |
| <b>Relationship</b>       |  |
| <b>Initiating</b>         | Actor (buyer, seller, banker)  |
| <b>Collaborating</b>      | DB Agent, Buyer Agent, Seller Agent, Banker Agent  |
| <b>Data Requirements</b>  |  |
| <b>Data Required:</b>     | Login name, password, username, Ad expiration, last login time   |

**Table 3 An example of role documentation after analysis.**

**Or we also can use following table to document the analysis results such as in Table 4:**

|                  |  |                                      |   |
|------------------|--|--------------------------------------|---|
| ROLE SCHEMA:     | CoffeeFiller   |                                      |   |
| DESCRIPTION:     | This role involves ensuring that coffee is kept filled, and informing the workers when fresh coffee has been brewed.   |                                      |   |
| PROTOCOLS:       | Fill, InformWorkers, CheckStock, AwaitEmpty  |                                      |   |
| PERMISSIONS:     | reads  | supplied coffeeMaker<br>coffeeStatus | // name of coffee maker<br>// full or empty |
|                  | changes  | coffeeStock                          | // stock level of coffee                    |
| RESPONSIBILITIES | <p>CoffeeFiller = (Fill, InformWorkers, CheckStock, AwaitEmpty)<sup>69</sup></p> <ul style="list-style-type: none"> <li>• <math>coffeeStock \geq 0</math></li> </ul> |                                      |   |

**Table 4: An example for role CoffeeFiller**

#### Step 4 Role model verification and validation

After role model has been properly documented, a verification and validation process is needed in order to make sure the result is valid:

- Verification means to find whether there exists some ambiguous role, ambiguous interactions or relationship within the documentation.
- Validation means to find whether the documented results are consistent with the initial requirements of the system or the modeled organization.

Overall, only verified and validated documentation can be submitted to the design or implementation teams. Otherwise, a proper rework is needed in order to get quality system.

## 5 Conclusions

Role modelling is an increasing interesting topic in both Object-oriented methodology and agent based software engineering. How to model the agent system effectively is the challenge work for all the researchers in Agent based software engineering. As any other software system design methodology, an effective analysis method for agent system is definitely very important for the system design and implementation. This tutorial provides a basic introduction of the role modelling methodology for the analysis of agent system.

It is worth to note that this paper is large based on the current research results in the Agent and OO research communities. And the introduction is only part of the current research results. There are very limited part belongs to the author's own idea. All related references are cited. And the author apologizes for any missing references if it has.

## Reference

- [1] Wooldridge M. J., Jennings N. R. and Kinny D. The Gaia methodology for agent-oriented analysis and design, *Autonomous Agents and Multi-Agent Systems*, 3(3):285–312, September 2000. Manufacturing processes.
- [2] Wooldridge M. J., Jenning N. R. and Kinny D, A methodology for agent-oriented analysis and design. In *Proc. of the third international conference on Autonomous Agents*, pages 69–76, 1999.
- [3] DeLoach S. A. Multiagent Systems Engineering A Methodology and Language for Designing Agent Systems, In *Proc. of Agent Oriented Information Systems*, 1999. Manufacturing processes.
- [4] Wood M. F. and DeLoach S. A, An Overview of the Multiagent Systems Engineering Methodology, the First International Workshop on Agent-Oriented Software Engineering (AOSE-2000), 2000.
- [5] E. A. Kendall, Role models-patterns of agent system analysis, *BT Technol Journal* Vol. 17, No 4 October 1999
- [6] Giacomo Cabri, Letizia Leonardi & Franco Zambonelli, Modeling Role-based Interactions for Agents, *Workshop: Agent-oriented methodologies, OOPSLA 2002*, SEATTLE, WA, USA, Nov, 4-8.
- [7] Y. Shoham, An Overview of Agent-Oriented Programming. In *Software Agents*, J. M. Bradshaw (Ed.), pp. 271-290. MIT Press, 1997.
- [8] Helm, R., I. M. Holland, D. Gangopadhyay, “Contracts: Specifying Behavioural Compositions in Object- Oriented Systems,” *Object-oriented Programming, Systems, and Languages, ECOOP/OOPSLA’90 Proceedings*, 1990, pp. 169 – 180.
- [9] E.A. Kendall. "Role Modelling for Agent System Analysis, Design, and Implementation," *International Conference on Agent Systems and Applications/ Mobile*
- [10] Riehle, D., "Bureaucracy", in *Pattern Languages of Program Design 3*, R. Martin, D. Riehle, F. Buschmann (Ed.), Addison Wesley, 1998, pp. 163 -185.
- [11] M. Wooldridge, N. R. Jennings, and D. Kinny, In O. Etzioni, J. P. Muller, and J. Bradshaw, editors: *Agents '99: Proceedings of the Third International Conference on Autonomous Agents* Seattle, WA, May 1998.
- [12] Behrouz Homayoun Far, *Agent Oriented Software Engineering*.