 UNIVERSITY OF CALGARY	Course Number: SENG 609.22	Course Name: Agent-based Software Engineering
	Session: Fall, 2003	Department: Electrical and Computer Engineering
		Document Type: Tutorial Report

Methodologies for Agent Based Analysis and Design

Kuljinder(Bobby) Khatkar
 University of Calgary
 bkhatkar@yahoo.ca

Abstract

This tutorial explores the methodologies that can be used for Agent based analysis and design and see why they are inadequate in describing agent based analysis and design and then it describes comprehensive GAIA[1] methodology.

1. Introduction

The progress in software engineering is made possible by the development of powerful natural abstractions by which complex systems can be modeled. We have seen abstractions in forms such as procedural abstraction, abstract data types and object abstraction. Agent based methodology represent another powerful way of abstracting complex distributed systems.

Looking at the different techniques available for agent analysis and design.

Object oriented analysis and design

We can extend and adapt existing object oriented analysis and design and use it to describe Agent analysis and design.

Knowledge based

We can build on existing modeling techniques from knowledge engineering.

Formal Methods

We can use formal methods and languages such as Z and provide definitions to support agent systems.

Looking at the present techniques of software analysis and design such as Object oriented analysis and design, would be unsuitable to address this task[1]. Present software Analysis and Design techniques fail to address Agent's autonomous problem solving behavior, interactions among subsystems, and complexity of agent's system organizational structure. Remaining section gives an overview of GAIA methodology for Agent based analysis and Design and do comparison of various techniques.

2. GAIA Methodology

GAIA methodology is applicable to the wide range of multi agent systems and in particular both micro level and macro level. The basis of this methodology is viewing system as a computational organization. The agents in the system are realized as the basic computational units in the organization, which plays different roles to realize the organization as a whole.

2.1 Applicable Domain characteristics:

The domain in which this methodology is based considers agents as coarse-grained computational systems, which uses significant computational resources. Agents are considered heterogeneous where different agents are implemented in different programming languages and the overall system has small number of agents.

2.2 Conceptual framework

This methodology allows analyst to go systematically from detailed requirements to a sufficiently detailed design, which can be implemented directly. As you go downward from system analysis model to detailed design, you are more biased towards implementation.

2.3 Analysis Concepts

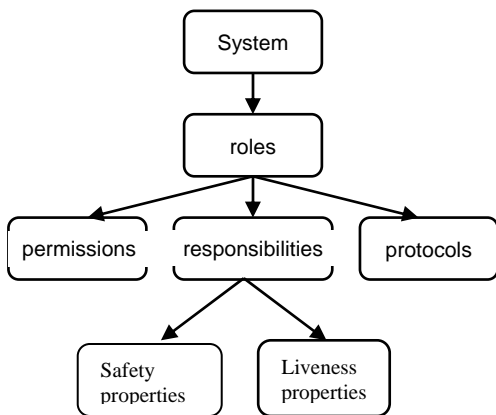


Fig 1 Analysis concepts

The root level abstract concept is a system. The system in this methodology means artificial society and organization. The next level in the hierarchy is roles. The analogy here is roles of staff members in the organization. The roles here will be instantiated actually by the individuals and the agent roles here is not necessarily static, it could be dynamic, for example one role can be taken by other agents. A role is defined by three attributes *permissions*, *responsibilities* and *protocols*. Responsibilities determine functionality and is a key attribute associated with a role. Responsibilities are of two types: *Liveness* and *Safety*. Liveness properties mean that something can be done and under what conditions will something be done. Safety properties on the other hand are invariants. Example of liveness properties is a Coffe Filler which has all properties – Fill, Inform workers, Check Stock, await empty and safety property is that coffee stock is never empty. Permissions are the rights associated with the role. A role is also associated with a number of protocols, which define the way that it can interact with other roles.

2.2 Methodology Models

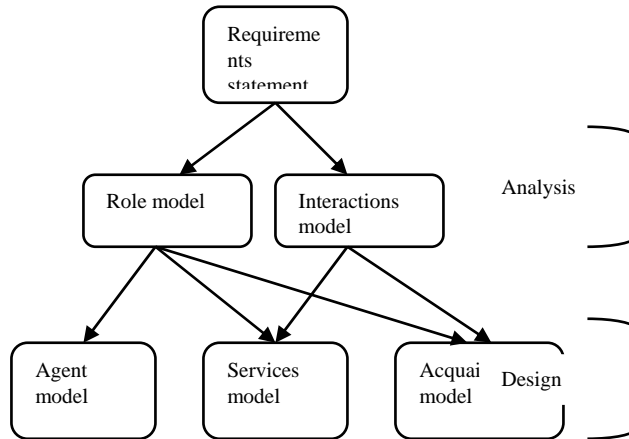


Fig 2: Relationships between methodology's models

Analysis Phase

The purpose of the analysis phase is to develop an understanding of the system and its structure as a multi-agent organization. This understanding is captured in the system organization.

Analysis phase consist of role model and interactions model. The *roles model* identifies the key roles in the system. Here the role can be viewed as an abstract description of an entity's expected function. The roles are characterized by *permissions* and *responsibilities*.

The *interactions model* consists of a set of protocol definitions, one for each of the inter-role interaction. The protocol here is an institutionalized pattern of interaction.

Analysis process in summary

- identifies the roles in the system.
Output: Output here is a prototypical roles model with a list of key roles of the system.
- For each role, identify and document the associated protocols.
Output: An interaction model, which has recurring patterns of inter-role interaction.
- Using the protocol model as a basis, roles model is elaborated with permissions and responsibilities.

Design Phase

The aim of the design process is to transform the abstract models derived during the analysis stage into models at a lower level of abstraction so that it can be implemented.

Design models consist of *Agent model*, *Service model* and *acquaintance model*.

Agent Model: This model identifies the agent types that will make up the system, and the agent instances that will be instantiated from these types.

Service Model: It identifies the main services that will be associated with each agent type.

Acquaintance Model: It documents the communication links that exist between agent types.

Design Process

A typical design process consist of

- Creating an agent model, which groups common roles under one agent type and form agent type hierarchy.
- Second step is to develop services model, by examining protocols and safety and liveness properties of roles.
- The third step is developing an acquaintance model from the interaction model and agent model.

3. Conclusion

This tutorial describes comprehensive approach called GAIA methodology for Agent based analysis and design. The key concepts in this methodology are roles, which have properties such as responsibilities, permissions and protocols. Roles can interact with one another in certain institutionalized ways, which are defined in the protocols of the respective roles.

4. References

1. Wooldridge, Michael, Nicholas R. Jennings, and David Kinny, "The Gaia Methodology for Agent-Oriented Analysis and Design,"
International Journal of Autonomous Agents and Multi-Agent Systems, 3:Forthcoming, 2000.