Chapter V

A Modeling Methodology for Intelligent Agents: An Electronic Commerce Application

Sooyong Park
Sogang University

Vijayan Sugumaran
Oakland University, USA

ABSTRACT

The unprecedented growth of the internet and web-based applications has necessitated the deployment of intelligent agents to facilitate some of the tedious and time-consuming activities on the Web. Consequently, there is an increasing demand for agent-oriented systems, which are soon becoming large and complex. To support a systematic development of such systems, an agent-oriented software development methodology is necessary. This chapter focuses on the modeling phase of agent-oriented software life cycle and, presents an approach for agent modeling consisting of Agent Elicitation, Intra, and Inter Agent modeling methods. Agent Elicitation deals with identifying and extracting agents from “classes” in the real world. Intra Agent Modeling involves expressing agent characteristics such as goal, belief, plan and capability, whereas, Inter Agent modeling incorporates agent mobility and communication.

INTRODUCTION

As web-based applications get large and complex, sophisticated technologies are needed to support and execute heterogeneous and distributed applications. To manage this complexity, intelligent agent technology is beginning to be employed as part of the solution in various web applications (Wooldridge, 1999). Since its introduction in the AI community, agent technology has permeated to various application domains as simple as e-mail filtering, to as complex as Air-traffic Control (Jennings, 1998). Recently, in distributed and heterogeneous environments such as Electronic Commerce (EC) applications, intelligent agents are increasingly being utilized to perform various tasks.

Since agents are used in many application areas, a systematic approach that is grounded within the software engineering paradigm is highly important for the development
of agent-oriented software. However, there has not been enough research on this subject in the Software Engineering Community.

The fundamental question is how to model software in an agent-oriented paradigm, similar to object-oriented paradigm, which utilizes procedural and data abstractions. To facilitate the design and development of agent-based systems, higher levels of abstractions are necessary to accurately model the flexible, dynamic, and autonomous problem solving characteristics of agents. Some of the agent behaviors may be application domain specific, while others are domain independent. This triggers several research questions such as how best to analyze and model the problem domain in order to facilitate agent-oriented software development, how to identify potential agents within the problem domain, how to model the internal and external behaviors of agents, the inter agent communication and cooperative problem solving, and agent mobility. Our research has been motivated by the above mentioned questions and attempts to develop an agent modeling methodology that facilitates capturing the domain independent and domain dependent aspects of agents. This chapter focuses on the initial phase of agent-oriented software development which includes: a) domain analysis – problem domain modeling as well as agent identification, and b) agent modeling consisting of intra agent and inter agent modeling.

We assume that the real world consists of agents and objects, and an agent is similar to an active object (Jennings, 1998) or a distributed object (Schroeder, 1999). In our agent-oriented process model, we obtain objects from problem domain analysis using UML (Unified Modeling Language) (Harmon, 1998; Selic, 1998), then, extract and create agents from these objects using agent selection rules. Typically, the agent modeling activity consists of two parts: intra agent modeling and inter agent modeling. The former focuses on agent’s attributes and behaviors resulting in an “Intra Agent Model,” whereas, the latter concentrates on agent communication (message exchanges) and mobility, yielding an “Inter Agent Model.”

The remainder of the chapter is organized as follows. Section 2 briefly discusses agent characteristics and modeling methods. Section 3 describes the first two phases of our lifecycle model, namely, domain analysis and agent modeling. Specifically, the UML based problem domain analysis, as well as the agent elicitation process based on Agent Selection Rules are described. This section also discusses the intra agent and inter agent modeling. We have applied our approach to a simple agent-based application in the electronic commerce domain, which is presented in section 4. We have also developed a proof-of-concept prototype to demonstrate the feasibility of our approach using the Zeus agent building toolkit (BT, 1998), which is discussed in section 5. Section 6 provides summary and future research.

**AGENT PROPERTIES AND AGENT MODELING METHODS**

The concept of agent was introduced by John McCarthy in the mid-1950’s and established by Oliver G. Selfridge several years later (Kay, 1984). In the early years, though many researchers investigated different aspects of the agent technology, it was still not considered as mainstream research within the AI community. However, since the late 80’s, there has been a resurgence of interest in agent technology, and currently we are seeing a proliferation of agent-based applications, particularly on the Web.

Though several characteristics of agents have been discussed in the literature (Jennings, 1998; Nwana, 1996), we have concluded that the following three properties are essential to
Collaboration Challenges in Community Telecommunication Networks
www.igi-global.com/article/collaboration-challenges-community-telecommunication-networks/2898?camid=4v1a