Chapter 17

From Object-Oriented Modeling to Agent-Oriented Modeling: An Electronic Commerce Application

Sooyong Park
Sogang University, Korea

Vijayan Sugumaran
Oakland University, USA

The use of intelligent agents is on the rise, fueled by the unprecedented growth in the internet and web based applications. Consequently, agent-oriented software is becoming large and complex. To support a systematic development of such software, 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 as Goal, Belief, Plan and Capability whereas, Inter Agent modeling incorporates agent mobility and communication in a multi-agent system.

INTRODUCTION

As software applications get large and complex, very sophisticated environments are needed to support and execute heterogeneous and distributed real-time
To manage this complexity, intelligent agent technology is beginning to be employed as part of the solution in various software environments (Woolridge, Jennings and Kinny, 1999). Since its introduction in the AI community, agent technology has permeated to various other applications such as e-mail filtering, Air-traffic Control etc (Jennings, Sycara and Wooldridge, 1998). Moreover, in distributed and heterogeneous environments such as Electronic Commerce (EC) applications, intelligent agents are increasingly 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.

To develop agent-oriented software, we introduce a process model, which is an adaptation of the traditional Waterfall model and contains the following major activities:

a) domain analysis - problem domain understanding and modeling, and agent identification,
b) agent modeling - intra-agent and inter-agent modeling,
c) agent design - agent architecture and componentization,
d) agent implementation – implementing the agents using appropriate agent building tools and agent communication languages,
e) agent integration – integration of multi-agents and other components, and
f) verification and validation – testing and simulation of the agent functionalities.

Based on our agent-oriented software process model, this paper focuses on the initial phases of the model, namely, domain analysis which includes problem domain modeling and agent identification, and agent modeling which includes intra and inter agent modeling.

We believe that the real world consists of agents and objects, and an agent is similar to an Active object (Jennings et al., 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) (Rao and Georgeff, 1991; Wooldridge, 1997), 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. The next section briefly discusses agent characteristics and modeling methods. The third section 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
Related Content

Blackout Avoidance and Energy Saving Services with Alert Aggregation

Modulation Recognition for Software Defined Radio Signal
[www.irma-international.org/chapter/modulation-recognition-software-defined-radio/66479/](www.irma-international.org/chapter/modulation-recognition-software-defined-radio/66479/)

Collaborative Development of Dependable Cyber-Physical Systems by Co-Modeling and Co-Simulation

Exploiting Codified User Task Knowledge to Discover Services at Design-Time
[www.irma-international.org/article/exploiting-codified-user-task-knowledge-to-discover-services-at-design-time/78917/](www.irma-international.org/article/exploiting-codified-user-task-knowledge-to-discover-services-at-design-time/78917/)

A Tool Support for Secure Software Integration
[www.irma-international.org/article/tool-support-secure-software-integration/43925/](www.irma-international.org/article/tool-support-secure-software-integration/43925/)