IDEA GROUP PUBLISHING **ITB13068**

701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

This paper appears in the publication, *Advanced Topics in Database Research, vol. 5*
edited by Keng Siau © 2006, Idea Group Inc.

Chapter IX

Externalisation and Adaptation of Multi-Agent System Behaviour

Liang Xiao, Queen's University Belfast, UK
Des Greer, Queen's University Belfast, UK

ABSTRACT

This chapter proposes the adaptive agent model (AAM) for agent-oriented system development. In AAM, requirements can be transformed into externalised business rules. These rules represent agent behaviours, and collaboration between agents using the rules can be modelled using extended UML diagrams. Specifically, a UML structural model and a behavioural model are employed. XML is used to further specify the rules. The XML-based rules are subsequently translated by the agents. The UML diagrams and XML specification can both be edited at any time, the newly specified behaviours being available to the agent system immediately. An illustrative example is used to show how AAM is deployed, demonstrating adaptation of inter-agent collaboration, intra-agent behaviours, and agent ontologies. With AAM, there is no need to recode and regenerate the agent system when change occurs. Rather, the system model is easily configured by users and agents will always get up-to-date rules to execute at run-time.

INTRODUCTION

Agent-oriented systems differ from object-oriented systems in that agents are active, while objects are passive. Thus, agents have the goal of having dynamic behaviours. Therefore, agent systems should be easily adaptable, being easily changed by engineers. Better still would be that they were adaptive, where systems change their behaviours according to their context (Lieberherr, 1995).

Although many tools and techniques are available for agent-oriented systems development, there is no unified and mature way to do it. What is more, existing agent platforms, like Java Agent Development (JADE) (Bellifemine, Caire, Poggi, & Rimassa, 2003), require designers and developers to code agent behaviours in fixed methods and the way to write them varies from one platform to another. This lack of uniformity of approach means that maintaining agent systems is potentially expensive. Being able to automatically generate agent systems and adapt their behaviours with changing requirements would alleviate this maintenance burden.

The objective of the chapter is to find a way to externalise agent behaviours in a repository. The configuration of the agent behaviours can be made at run-time by changing the repository, supported by tools. Therefore, new requirements can be continually reflected in the agent systems. We call this repository a requirements database and our approach adaptive agent model. The requirements database is in XML format, and the stored agent behaviours are represented as business rules.

BACKGROUND

In this section, we will first briefly introduce agent systems in general, and discuss how such systems are currently developed. We then present some existing approaches towards system adaptivity. After that, business rules, being able to capture system behaviours, are presented as a means to achieve more flexible agent behaviour code. Following the demonstration of possible implementation of rules as agent behaviours, we come back to the design aspect and describe the addition of the rule in two existing extended UML notation systems, their usefulness, and insufficiencies. Finally, our perspective and the main idea of our approach are given.

Agent Systems and Platforms

Software agents are defined as follows: "An agent is an encapsulated computer system that is situated in some environment, and that is capable of flexible, autonomous action in that environment in order to meet its design objectives" (Jennings, 2000, p. 280). Sending and receiving messages are the two main activities of agents. Various agent system development platforms are available, the JADE framework being one of them. JADE is aimed at developing multi-agent systems and applications conforming to Foundation for Intelligent Physical Agents (FIPA) (2005) standards. With JADE, an agent is able to carry out several concurrent tasks in response to different external events. To date, developers have, generally, been required to write repetitive and tedious code for the behaviour of every agent manually.