Chapter 2.12
An Architecture for Learning Environments Based on the Lightweight Integration of Intelligent Agents

Sergey Butakov
SolBridge International School of Business, South Korea

Denis Smoline
American University of Nigeria, Nigeria

Nousheen Naimat Samuel
SolBridge International School of Business, South Korea

Naima Naimat Samuel
SolBridge International School of Business, South Korea

ABSTRACT

The main objectives of this chapter are to provide a review of multi-agent platforms for learning environments, to discuss the use of lightweight integration technologies in learning management systems, and to outline the theoretical framework for a multi-agent based platform for course provisioning. The proposed framework places a message switching system at the center of the architecture, providing an effective yet lightweight mechanism for agent communications. The chapter also provides two practical examples of the implementation of lightweight integration technologies for course management system development. The examples show the feasibility of implementing the proposed lightweight architecture, showing how the agent based elements could be added to existing learning management systems.

DOI: 10.4018/978-1-4666-0011-9.ch2.12

INTRODUCTION TO MULTI-AGENT AND LIGHTWEIGHT SYSTEMS

Multi-agent environments, or multi-agent systems (MAS), were initially introduced in the Artificial Intelligence (AI) area in 1980s. In the computer science area, actor/agent based systems have been defined as sets of independent computational units/agents with a finite number of states and possible outcomes of the actions, where the agents can act independently and asynchronously, communicating either on a peer-to-peer basis or through a shared memory (Agha, 1986). It is notable that researchers from AI have emphasized anthropomorphic characteristics of the agents, such as intelligence, intentions, and knowledge (Woodridge & Jennings, 1995).

From the software developer's perspective, there are two major components that play critical roles in MAS: the agents themselves and the environment that facilitates their communications within the system and with the outside world. According to Woodridge and Jennings (1995), a single agent should have the following properties: autonomy, social ability, reactivity, and pro-activeness. The environment should facilitate communications within the system, employing either system calls or communication protocols and languages (Genesereth & Ketchpel, 1994), and also maintain interfaces to the outside world.

According to the exhaustive review by Woodridge and Jennings (1995), MAS did not enter the mainstream of research in AI and computer science until the mid to early. In the beginning of the 1990s, researchers from the software engineering area considered MAS as one of the possible approaches to improve the scalability and reliability of large software applications and employ theoretical concepts from AI in mainstream software systems. At the same time, there was also significant effort towards the standardization of communication languages for agent environments. Two major approaches were used to build such languages – the procedural approach and declarative approach (Genesereth & Ketchpel, 1994). As shown in the review by Flores-Mendez (1999), the two most successful efforts at that time were the Knowledge Query and Manipulation Language, or KQML, and ontology-based languages. As the concept of MAS became progressively more popular in software engineering, many efforts were pursued in the area of MAS architecture standardization. Florez-Mendez indicated the following efforts as the most successful in this area: the model developed by the Object Management Group, model developed by the Foundation for Intelligent Physical Agents, the KAoS system and the model developed by General Magic. For more information on these models, the reader can refer to Flores-Mendez's review (Flores-Mendez, 1999).

A new wave of interest in MAS came with the development of the internet and the introduction of web services in the early 2000s. At that time, agent-based technologies diffused into a new area of internet based systems. For example, Gibbins and coauthors proposed to enrich web services with agent-like functionality and ontological descriptions in order to improve communications in web services based systems (Gibbins, Harris and Shadbolt, 2003).

Of course, Learning Management Systems (LMS) were not immune to such a major trend in software development practices. There are a number of works aimed at the development of education-related systems based on MAS principles. Han et al. proposed an integration infrastructure that allows linking together agents from e-learning and e-library platforms. They listed the following agents as the major actors within such a system: curriculum management agents, content management agents, security agents, and digital library supervision agents (Han & Han, 2005). Of course, this list is not exhaustive and should be extended in the general framework for an MAS based learning environment.

Zaka and Safran (2008) also noted the limited collaboration features of the most popular e-learning management platforms. They empha-