Chapter I
Toward Agent-Oriented Conceptualization and Implementation

Pratik K. Biswas
Avaya Inc., USA

ABSTRACT

The desire to flexibly customize software, manage it efficiently, and empower it with intelligence has driven research and development-related efforts toward intelligent agents. The benefits in terms of rapid delivery, reduced costs, and enhanced productivity can be realized in the areas of systems and software engineering with the proliferation of this technology. Intelligent agents represent an alternate approach to distributed software engineering. Agent-oriented conceptualization provides a new paradigm for the design and development of these agent-based systems. This chapter extends and formalizes this agent-oriented modeling approach to the conceptualization process. It defines agent models and proposes a high-level methodology for agent-oriented analysis and design. It also looks at the heart of agent-oriented programming and outlines its advantages over traditional approaches to distributed computing and interoperability. The chapter includes analogies with the object-oriented methodologies and other existing agent-oriented methodologies wherever applicable. It reviews the Foundation of Intelligent Physical Agents-compliant infrastructure for building agent-based systems and suggests a multi-agent systems framework that merges this infrastructure with the emerging J2EE technologies. The chapter concludes with a case study and an insight to future challenges.
INTRODUCTION

Agent-based computing represents a novel software engineering paradigm that has emerged from merging two technologies (Odell & Burkhart, 1998), namely artificial intelligence (AI) and object-oriented distributed computing. Agent-based systems aim to strike a balance between artificial intelligence and computational utility.

Agents are intelligent, autonomous, software components capable of interacting with others within an application, attaining a common goal and thereby contributing to the resolution of some given problem. They are important because they inter-operate within modern applications like electronic commerce and information retrieval. Multi-agent systems (Biswas, 2005; Brazier, Keplicz, Jennings, & Treur, 1995; Brazier, Dunin-Keplicz, Jennings, & Treur, 1997; Jennings, 2001; Lind, 2000) are composed of a set of agents and are useful for the modeling and development of distributed information systems with synchronous or asynchronous component interactions. Multi-agent systems (MAS) differ from non-agent-based systems because agents are intended to be autonomous units of intelligent functionality who can interact with others through high-level protocols and languages.

A system can be successfully built and deployed if it has been properly conceptualized. Conceptualization requires an appropriate set of abstractions and a methodology for system specification, analysis, and design. However, many of our traditional ways of thinking about and designing software do not fit the multi-agent paradigm. In this chapter, we advocate an agent-oriented paradigm for conceptualizing the analysis and design of agent-based systems. The remainder of the chapter is organized as follows. In the second section, we provide motivations for this chapter. In the third section, we take a more detailed look at intelligent agents, their characteristics, capabilities and interactions. In the third section, we introduce the concept of agent-oriented thinking. In the fourth section, we discuss agent-oriented modeling, define a new agent model, describe the steps involved in our approaches to agent-oriented analysis and design, and compare them with object-oriented methodologies. In the fifth section, we discuss agent communication and mobility, at the heart of agent-oriented programming, outline their advantages over traditional approaches to distributed computing, compare agent-oriented programming with object-oriented programming, and show why Java is the natural choice for its implementation. In the sixth section, we present a Foundation of Intelligent Physical Agents (FIPA)-compliant platform for building agent-based systems and then suggest a multi-agent systems framework using the J2EE technologies that can implement this platform. In the seventh section, we present a case study to illustrate the applicability of our methodology. In the next section, we compare our methodology with other existing agent-oriented methodologies. Finally, in the last section, we summarize our results and provide an insight to the future challenges. This chapter is an extension of the work presented in Biswas (2007).

MOTIVATIONS

Over the years, a wide range of software engineering paradigms have been devised (e.g., procedural programming, structured programming, declarative programming, object-oriented programming, design patterns, application frameworks, and component-ware) to deal with the increasing complexity of software applications. Although each successive development claims to make the engineering process easier, researchers continually strive for more efficient and powerful software engineering techniques, especially as solutions for ever more demanding applications are required.

Most real-world applications of today are significantly more complex than before as they