Chapter 6

Building an Agent: By Example

Paul Darbyshire
School of Information Systems
Victoria University of Technology, Australia

Since the emergence of agent technology, there have been many papers and articles written on the advantages and use of the technology. In particular, in the last two years, the number of papers discussing the use of agent systems seems to have risen exponentially. Whether this rise in the interest of agent technology corresponds to the emergence of eCommerce, Internet banking, and the explosion in Web-based systems, or the maturity of the technology and programming languages used to develop them is another matter. For people interested in the technology and wanting to build their own agents, most of the material provides little insight in how to actually build an agent. This chapter discusses the problem of actually building an agent using an example of an “email helper” agent.

INTRODUCTION

Over the last decade, the programming languages and the maturity of the support technology used to build agents has developed to a level where agents have migrated from the laboratory to real-world applications. In particular, over this time, the Web has gained prominence, and the number of Web-based applications has grown enormously. The emergence of Web-based education, eCommerce, and eBusiness-type applications have coincided with the rise in development and popularity of the Web. The ensuing activity has generated enormous interest in agent technology and solutions. This popularity and interest in agent technology is also reflected in the number of papers devoted to agent-based systems. However, for the novice trying to understand and develop his or her
own agents, the majority of papers and articles offer little insight as to the architecture and framework used for building such systems.

The purpose of this chapter is to outline the process of building an agent, and this is done through an example of an email helper agent. The example uses the Java language to develop an architecture for building an agent, and a communication framework in which the agents operate. Another agent called an interface agent is outlined that can be used as a generic interface agent between a human operator, other agents, and the agent environment. The agents developed in this chapter are not fully functional but do give enough information to enable the novice developer to get started with some basic agent coding.

In the following sections, a brief background on agents is given, followed by a treatment on the selection of Java as the development language. A framework for the environment in which the agents operate is described, and then a general simple architecture for building an agent is shown. Finally, an interface agent is shown to allow the user to interact with other agents and the environment.

BACKGROUND

Before we begin to develop agents we really must be in a position to define what an agent is. This in itself is not an easy task, as no universally acknowledged definition of a software agent is currently available. Even the “experts” cannot agree on a definition for an agent (Nwana, 1996; Wooldridge & Jennings, 1995). Although the concept of an agent has been around for some time, agent software is still an emerging technology that can still be regarded as being in an embryonic stage. Despite this, the range of organizations and disciplines researching and pursuing agent technology is broad. The term “agent” is being increasingly used by computer researchers, software developers and even the average computer user, yet when pressed, many would be unable to give a satisfactory explanation of just what an agent really is.

Agent technology emerged from the field of AI research, so the term “Intelligent Agent” is often used. However, agents need not be intelligent, and in fact most tasks do not warrant the use of “smart agents” (Nwana, 1996). Other adjectives often used with agents are: interface, autonomous, mobile, Internet, information and reactive. The term “agent” can be thought of as an umbrella under which many software applications may fall, but is in danger of becoming a noise term due to overuse (Wooldridge & Jennings, 1995). Many agents are currently characterized by descriptive terms that accompany them, for example intelligent, smart, autonomous.

What makes agents different from standard software is the characteristics that agents must possess in order to be termed such. There are a number of classification schemes that can be used to typecast existing agents, and these in-
Related Content

**Advances in FCA-based Applications for Social Networks Analysis**
[www.igi-global.com/article/advances-in-fca-based-applications-for-social-networks-analysis/80383?camid=4v1a](www.igi-global.com/article/advances-in-fca-based-applications-for-social-networks-analysis/80383?camid=4v1a)

**An Ontology-Based Context-Aware Infrastructure for Smart Homes**
Bin Guo, Daqing Zhang and Michita Imai (2011). *Handbook of Research on Ambient Intelligence and Smart Environments: Trends and Perspectives* (pp. 272-299).
[www.igi-global.com/chapter/ontology-based-context-aware-infrastructure/54663?camid=4v1a](www.igi-global.com/chapter/ontology-based-context-aware-infrastructure/54663?camid=4v1a)

**System Identification Based on Dynamical Training for Recurrent Interval Type-2 Fuzzy Neural Network**
Tsung-Chih Lin, Yi-Ming Chang and Tun-Yuan Lee (2013). *Contemporary Theory and Pragmatic Approaches in Fuzzy Computing Utilization* (pp. 188-205).
[www.igi-global.com/chapter/system-identification-based-dynamical-training/67490?camid=4v1a](www.igi-global.com/chapter/system-identification-based-dynamical-training/67490?camid=4v1a)