Embedded Agents for Mobile Services

John F. Bradley
University College Dublin, Ireland

Conor Muldoon
University College Dublin, Ireland

Gregory M. P. O'Hare
University College Dublin, Ireland

Michael J. O'Grady
University College Dublin, Ireland

INTRODUCTION

A significant rise in the use of mobile computing technologies has been witnessed in recent years. Various interpretations of the mobile computing paradigm, for example, ubiquitous and pervasive computing (Weiser, 1991) and more recently, ambient intelligence (Aarts & Marzano, 2003)—have been the subject of much research. The vision of mobile computing is often held as one of "smart" devices operating seamlessly and dynamically, forming ad-hoc networks with other related devices, and presenting the user with a truly ubiquitous intelligent environment. This vision offers many similarities with the concept of distributed artificial intelligence where autonomous entities, known as agents, interact with one another forming ad-hoc alliances, and working both reactively and proactively to achieve individual and common objectives.

This article will focus on the current state of the art in the deployment of multi-agent systems on mobile devices and smart phones. A number of platforms will be described, along with some practical issues concerning the deployment of agents in mobile applications.

BACKGROUND

In the most general terms, an agent is one entity that acts, or has the authority to act, on behalf of another. In terms of information technology, an agent is a computational entity that acts on behalf of a human user, software entity, or another agent. Agents have a number of attributes that distinguish them from other software (Bradshaw, 1997; Etzioni & Weld, 1995; Franklin & Graesser, 1996; Wooldridge & Jennings, 1995):

- **Autonomy:** The ability to operate without the direct intervention from any entity, and possess control over their own actions and internal state.
- **Reactivity:** The ability to perceive their environment and react to changes in an appropriate fashion.
- **Proactivity:** The ability to exhibit goal-directed behavior by taking the initiative.
- **Inferential Capability:** The ability to make decisions based on current knowledge of self, environment, and general goals.
- **Social Ability:** The ability to collaborate and communicate with other entities.
- **Temporal Persistence:** The ability to have attributes like identity and internal state to continue over time.
- **Personality:** The ability to demonstrate the attributes of a believable character.
- **Mobility:** The ability to migrate self, either proactively or reactively, from one host device to another.
- **Adaptivity:** The ability to change based on experience.

An agent requires some space where it can exist and function, and this is provided for by an agent platform (AP). An AP comprises "the machine(s), operating system, agent support software,…agent management components…and agents" (FIPA, 2000, p. 6). The AP allows for agent creation, execution, and communication.

The majority of computer systems currently in operation use algorithms that are based on the concept of perfect information. The problem is that in the real world, businesses often require software functionality that is much more complex than this (Georgeff, Pell, Pollack, Tambe, & Wooldridge, 1999). Typically, computational entities within these systems should have an innate ability to deal with partial information and uncertainty within their environment. These types of systems are highly complex and are intractable using traditional approaches to software development. The rate at which business systems must change, due to market pressures and new information coming to light, requires software architectures and languages that more efficiently

manage the complexity that results from alterations being made to the code and the specifications.

Agent architectures, and in particular belief-desire-intention (BDI) (Rao & Geogeff, 1995) agent architectures, are specifically designed to deal with these types of issues and thus contain mechanisms for dealing with uncertainty and change. A problem with traditional systems is that they assume that they exist within a static or constant world that contains perfect information. The types of mobile systems that we are concerned with are dynamic and perhaps even chaotic, embedded with agents that have a partial view of the world and which are resource bounded.

Agents rarely exist in isolation, but usually form a coalition of agents in what is termed a multi-agent system (MAS). Though endowed with particular responsibilities, each individual agent collaborates with other agents to fulfill the objectives of the MAS. Fundamental to this collaboration is the existence of an Agent Communications Language (ACL), which is shared and understood by all agents. The necessity to support inter-agent communication has led to the development of an international ACL standard, which has been ratified by the Foundation for Intelligent Physical Agents (FIPA).

JAVA 2 MICRO EDITION (J2ME)

Most agent platforms developed for mobile devices have been written in the Java programming language—on mobile devices that usually means Java 2 Micro Edition (J2ME). This edition of Java contains a cut down API, a reduced footprint Java Virtual Machine, and a slightly different syntax (e.g., parameterized classes in Java 5). Java applications that contain dependencies on the idiosyncrasies of the different editions cannot be ported to a different range of devices without making alterations to the code. Their performance, however, is improved because the code is no longer developed to the lowest common denominator. Different algorithms and coding styles are now used for desktop machines and embedded devices rather than adopting comprised or overarching approaches that do not maximize the performance or maintainability of either.

A NUMBER OF AGENT PLATFORMS EXISTS FOR MOBILE DEVICES

3APL-M

3APL-M (Koch, 2005) is a platform that enables the fabrication of agents using the Artificial Autonomous Agents Programming Language (3APL) (Dastani, Riemsdijk, Dignum, & Meye, 2003) for internal knowledge representation. Its binary version is distributed in J2ME and J2SE compilations. 3APL provides programming constructs for implementing agents' beliefs, goals, basic capabilities, and a set of practical reasoning rules. The framework comprises an API that allows a Java application to call 3APL logic and deliberation structures.

Agent Factory Micro Edition

Agent Factory Micro Edition (AFME) (Muldoon, O'Hare, Collier, & O'Grady, 2006) is an agent platform developed for the construction of lightweight intelligent agents for cellular digital mobile phones and other compatible mobile devices. AFME is broadly based on Agent Factory (Collier, 2001), a pre-existing J2SE framework for the fabrication and deployment of agents. AFME differs from the original version of the system in that it has been designed to operate on top of the Constrained Limited Device Configuration (CLDC) Java platform augmented with the Mobile Information Device Profile (MIDP). CLDC and MIDP form a subset of the J2ME platform specifications. Though sharing the same broad objectives of the other projects mentioned in this section, AFME differs in a number of ways. With a jar size of 85k, it is probably the smallest footprint FIPA-compliant deliberative agent platform in the world. The platform supports the development of a type of software agent that is: autonomous, situated, socially able, intentional, rational, and mobile. An agent-oriented programming language and interpreter facilitate the expression of an agent's behavior through the formal notions of belief and commitment. This approach is consistent with a BDI agent model.

LEAP

Probably the most widely known agent platform for resource-constrained devices is the Light Extensible Agent Platform (LEAP) (Berger, Rusitschka, Toropov, Watzke, & Schichte, 2002). LEAP is a FIPA-compliant agent platform developed to be capable of operating on both fixed and mobile devices with various operating systems in wired or wireless networks. Since version 3.0, LEAP extends the Java Agent DEvelopment Framework (JADE) (Bellifemine, Caire, Poggi, & Rimassa, 2003) by using a set of profiles that allow it to be configured for various Java Virtual Machines (JVMs). The architecture of the platform is modular and contains components for managing the lifecycle of the agents and controlling the array of communication protocols. The platform is split into several agent containers, one for every device used. These containers are responsible for passing messages between agents and choosing the appropriate communication protocol. One of these containers, known as the main container, includes agents that fulfill the white and yellow pages services as necessitated by the FIPA specification.