Chapter 3
Developing Smart Emergency Applications with Multi-Agent Systems

Federico Bergenti
Università degli Studi di Parma, Italy

Agostino Poggi
Università degli Studi di Parma, Italy

ABSTRACT

Multi-agent systems have been importantly contributing to the development of the theory and the practice of complex distributed systems and, in particular, they have shown their potential to meet critical needs in high-speed, mission-critical, content-rich, distributed information applications where mutual interdependencies, dynamic environments, uncertainty, and sophisticated control play a remarkable role. Therefore, multi-agent systems are considered a suitable technology for the realization of e-health applications where the use of loosely coupled and heterogeneous components, the dynamic and distributed management of data, and the remote collaboration among users are often the most relevant requirements. This paper describes some of the main reasons why multi-agent systems are today considered one of the best technologies for the realization and deployment of advances for e-health applications and, in particular, of smart emergency applications. After an introduction on the inherent characteristics of the use of multi-agent systems for e-health, the paper presents the results of EU-scale project CASCOM: a real multi-agent system for the execution of smart emergency tasks.

INTRODUCTION

Multi-agent systems are one of the most interesting areas in software research and they have been importantly contributing to the development of the theory and the practice of complex distributed systems (see, e.g., Bordini et al., 2005). In particular, multi-agent systems have shown the potential to meet critical needs in high-speed, mission-critical, content-rich, distributed information systems where mutual interdependencies, dynamic environments, uncertainty, and sophisticated control play a singular role (Gasser, 2001).

DOI: 10.4018/978-1-4666-0909-9.ch003

Agent and *multi-agent system* are buzzwords that found their way into a number of technologies and they have been largely used, e.g., in Artificial Intelligence, Databases, Operating Systems and Computer Networks. Although there is no such thing as an accepted definition of an agent (see, e.g., Genesereth & Ketchpel, 1994; Wooldridge & Jennings, 1995), all proposed definitions agree that an agent is essentially an autonomous software entity that provides an interoperable interface and that behaves like a rational actor working on behalf of some client in pursuit of its own agenda. Agents are designed to operate in dynamic and uncertain environments, taking complex decisions at run-time, and the learning capabilities of some agents make them capable to improve their performances over time, thus avoiding repeated negative conditions and persisting on successful behaviours.

Even if a complex system can be realized in terms of a solitary agent working within its environment–that may or may not comprise users–usually, agent-based systems are made of multiple, interacting agents: agent-based systems are normally multi-agent systems. Multi-agent systems are generally considered an appropriate means for modelling complex, distributed systems, even if such a multiplicity naturally introduces the possibility of having different agents with potentially conflicting goals. Agents in a multi-agent system may decide to cooperate for mutual benefit, or they may compete to serve their own interests. They may take advantage of their social ability to exhibit flexible coordination behaviours that make them able to cooperate in the achievement of shared goals or to compete on the acquisition of resources and tasks. Finally, agents in a multi-agent system have the ability of coordinating their behaviours into coherent global actions. Coordination among agents is handled by means of a variety of approaches including negotiation (Jennings, 2001), organizational structuring (Horling & Lesser, 2005) and multi-agent planning (Durfee, 1999).

The very fact of characterizing agents in terms of the properties that they exhibit rather than in terms of a crude and often inapplicable definition, allows a plethora of software systems to be considered as first-class multi-agent systems. This makes multi-agent systems much more than a single technology supporting the realization of complex distributed systems. Multi-agent systems are *abstractions* capable of capturing the essence of many software systems at different levels of detail. In particular, agents and multi-agent systems are often considered the highest system level (Jennings, 2000) of today computing systems and they are meant to provide a truly novel level of abstraction in the analysis, design and implementation of complex software systems (Bergenti & Huhns, 2004). This is the reason why we can correctly account many recent software systems as multi-agent systems, even if no agent-based development technology were adopted in their realization. The agent-based nature of a system comes from the characteristics of its components and of the interactions among them, rather than from a hypothetical "agent based" label attached on the box of the development tool adopted for their realization. Notably, multi-agent systems are often developed using technologies that have no built-in notion of agent.

A lot of work has been done in the last decade for spreading the use of multi-agent systems for the realization of real-world software applications and services. Several technological specifications are the results of such a work. Among them, the two main results to date are: *(i)* FIPA specifications (FIPA, 2009), a set of specifications intended to support the interoperability between heterogeneous agent-based systems; and *(ii)* the JADE (Bellifemine et al., 2008; JADE, 2009) development framework that implements FIPA specifications and that supports interoperability between agents using consolidated technologies to provide for a transparent and dynamic allocation of fixed and mobile users and agents (Bergenti et al., 2001).