Chapter 9
An Advanced IDE for Designing Transparent Fuzzy Agents

Giovanni Acampora
University of Salerno, Italy

Enrico Fischetti
University of Salerno, Italy

Antonio Gisolfi
University of Salerno, Italy

Vincenzo Loia
University of Salerno, Italy

ABSTRACT

Recently, computational agents received significant attention in computer science research community. In fact, intelligent agents methodology is a powerful artificial intelligence technology showing considerable promise as a new paradigm for mainstream software development and able to offer new ways of abstraction, decomposition, and organization that fit well with our natural view of the world. However, despite their promise, intelligent agents are still scarce in the market place. A key reason for this is that developing intelligent agent software requires significant training and skill. Artificial Intelligence methodologies and computer networking tools represent the necessary basic knowledge to design and implement advanced agents oriented systems. This paper introduces an integrated development environment supporting the agents developers to design fuzzy-based agents in a simple and fast way. Proposed framework has been realized by integration of theoretical methodologies as fuzzy logic and labeled tree, together with Open Source Software tools as JaxMe2.

INTRODUCTION

Multi-Agent systems technology has generated lots of excitement in recent years because of its promise as a new paradigm for conceptualizing, designing, and implementing advanced software systems. This promise is particularly attractive for creating software that operates in environments that are distributed

DOI: 10.4018/978-1-61520-849-4.ch009

and open. More precisely, MASs consist of a large number of computational agents, interconnected by communication devices, that seamlessly work together in order to achieve prefixed common goals. The term *agent* describes a software entities characterized by the following features: *persistence*, *autonomy*, *social ability*, *reactivity*. Different methodologies and techniques have to be exploited in order to design advanced *multi-agent systems*, ranging from social science, business models, network architectures, up to human interaction design. This methodologies integration allows to design and implement agents-based environment characterized by following features:

- Whereas a centralized system may be plagued by resource limitations, performance bottlenecks, or critical failures, a MAS is decentralized and thus does not suffer from the "single point of failure" problem associated with centralized systems;
- A MAS allows for the interconnection and interoperation of multiple existing legacy systems. By building an agent wrapper around such systems, they can be incorporated into an agent society;
- A MAS models problems in terms of autonomous interacting component-agents, which is proving to be a more natural way of representing task allocation, team planning, user preferences, open environments, and so on;
- A MAS efficiently retrieves, filters, and globally coordinates information from sources that are spatially distributed;
- A MAS provides solutions in situations where expertise is spatially and temporally distributed;
- A MAS enhances overall system performance, specifically along the dimensions of computational efficiency, reliability, extensibility, robustness, maintainability, responsiveness, flexibility, and reuse.

These characteristics enhance performance along the dimensions of (1) *computational efficiency* because concurrency of computation is exploited (as long as communication is kept minimal, for example, by transmitting high level information and results rather than lowlevel data); (2) *reliability*, that is, graceful recovery of component failures, because agents with redundant capabilities or appropriate interagent coordination are found dynamically (for example, taking up responsibilities of agents that fail); (3) *extensibility* because the number and the capabilities of agents working on a problem can be altered; (4) *robustness*, the system ability to tolerate uncertainty, because suitable information is exchanged among agents; (5) *maintainability* because a system composed of multiple components-agents is easier to maintain because of its modularity; (6) *responsiveness* because modularity can handle anomalies locally, not propagate them to the whole system; (7) *flexibility* because agents with different abilities can adaptively organize to solve the current problem; and (8) *reuse* because functionally specific agents can be reused in different agent teams to solve different problems.

However, in spite of hard exploitation of computational agents in several domains of computer applications, MAS developers need of significant training, skills and experiences to design and develop advanced MAS-based frameworks. In fact, as previously depicted, MAS can be considered as a composition of different computer science backgrounds, mainly, *Artificial Intelligence* and *Computer Networking*. The main contribution of proposed paper is that it introduces an easy to use and expressive environment (FuzzyIDE) which allows *multi-agent system* designers to focus on the conceptual design and specification of their system (rather than having to worry about low-level system programming issues). This environment exploits fuzzy logic to define agents' intelligence through a collection of fuzzy rules used by agents to propose an advanced behaviour characterized by autonomy and reactivity. The