Chapter VI

Multi-Agent Systems as Computational Organizations: The Gaia Methodology

Franco Zambonelli
Università di Modena e Reggio Emilia, Italy

Nicholas R. Jennings
University of Southampton, UK

Michael Wooldridge
University of Liverpool, UK

Abstract

The multi-agent system paradigm introduces a number of new design/development issues when compared with more traditional approaches to software development and calls for the adoption of new software engineering abstractions. To this end, in this chapter, we elaborate on the potential of analyzing and architecting complex multi-agent systems in terms of computational organizations. Specifically, we identify the appropriate organizational abstractions that are central to the analysis and design of such systems, discuss their role and importance, and show how such abstractions are exploited in the context of the Gaia methodology for multi-agent systems development.
Introduction

In the past few years, multi-agent systems (MASs) have been recognized as an effective software engineering paradigm for designing and developing complex software systems (Jennings, 2001; Wooldridge & Jennings, 1995). In fact, the key characteristics of MASs (i.e., autonomy, situatedness, proactivity, and sociality [Wooldridge & Jennings, 1995]) are well suited to tackling the emerging complexities of modern software scenarios for a number of reasons. Firstly, the autonomy of the application components (i.e., the ability for an agent to decide what actions it should take at what time [Wooldridge & Jennings, 1995]) reflects the decentralized nature of modern distributed systems (Tennenhouse, 2000) and can be considered as the natural extension to the notions of modularity and encapsulation for systems that are owned by different stakeholders (Parunak, 1997). Secondly, the flexible way in which agents operate (balancing reactive behaviour in response to the environment in which they are situated, with proactive behaviour towards the achievement of their design objectives [Wooldridge & Jennings, 1995]) is suited to the dynamic and unpredictable situations in which software is now expected to operate (Zambonelli, Jennings, Omicini, & Wooldridge, 2001). Finally, the high-level, dynamic, and social nature of multi-agent interactions is appropriate to open systems in which the constituent components and their interaction patterns constantly change (Estrin, Culler, Pister, & Sukjatme, 2002).

In this chapter, we firstly elaborate on the organizational metaphor and present and discuss the various organizational abstractions that come into play in the design and development of a complex MAS. In particular, we show that an organization is more than simply a collection of roles (as most methodologies assume) and that, in order to effectively build an MAS in organizational terms, a variety of additional organization-oriented abstractions needs to be devised and placed in the context of a methodology. In addition, we discuss how the presented organizational abstractions are effectively and coherently exploited in the Gaia methodology. Specifically, the Gaia methodology (fully described in Zambonelli, Jennings, & Wooldridge, 2003) both promotes a simple and clear to manage the development process and exploits the appropriate organizational abstractions in the process of developing MASs.

The chapter is organized as follows. The next section introduces the organizational metaphor and describes the organizational abstractions that are to be exploited in agent-oriented software engineering. Following that, we detail how these abstractions are exploited in Gaia in order to provide a methodology for the analysis and design of MASs. The next section clarifies these concepts with the use of a simple case study, followed by a discussion of the weaknesses and strengths of the Gaia methodology. The chapter ends with our conclusions.