Chapter VIII

The MESSAGE Methodology for Agent-Oriented Analysis and Design

Francisco J. Garijo
Telefónica I+D, Spain

Jorge J. Gómez-Sanz
Universidad Complutense de Madrid, Spain

Philippe Massonet
CETIC, Belgium

Abstract

This chapter presents the MESSAGE agent-oriented software engineering methodology and illustrates it in an analysis and design case study. The methodology covers MAS analysis and design and is intended for use in mainstream software engineering departments. MESSAGE integrates into a coherent AOSE methodology basic agent-related concepts (such as organisation, role, goal, interaction, and task) that have so far been studied largely in isolation. The MESSAGE notation extends the UML with agent knowledge-level concepts and diagrams with notations for viewing them. The proposed diagrams extend UML class and activity diagrams. The
MESSAGE analysis and design process is based on the Rational Unified Process (RUP). The methodology distinguishes high-level from detailed design. An organisation-driven approach is presented for detailed design, where the global architecture of the MAS is derived from the structure and behaviour of the organisations that interact. Experimentation with the methodology shows that it supports the analysis and design of multi-agent systems that are flexible and adapt to specific kind of changes in a heterogeneous and dynamic environment.

Introduction

The agent-oriented (AO) approach promises the ability to construct flexible systems with complex and sophisticated behaviour by combining highly modular components. When several of these components interact, the intelligence of these components—the agents—and their capacity for social interaction results in a multi-agent system (MAS) with capabilities beyond those of a simple “sum” of the capabilities of the individual agents.

The availability of agent-oriented software engineering (AOSE) methodologies is a pre-requisite for engineering commercial MASs. The MESSAGE methodology covers the MAS analysis and design phases of the software engineering lifecycle. It is designed for use in mainstream software engineering departments that develop complex distributed applications. It is a genuinely agent-oriented methodology, but also builds upon the achievements of software engineering (SE), and is consistent with current SE best practice. MESSAGE grounds agent-oriented concepts in the same underlying semantic framework as that used by UML, and uses UML-based notation whenever appropriate.

MESSAGE (Caire, Courlier et al., 2001, 2002) extends UML by contributing agent knowledge-level concepts, such as organization (Zambonelli, Jennings, & Wooldridge, 2001), role (Kendall, 2000), goal (Dardenne, Lamsweerde, & Fickas, 1993; Giorgini, Kolp, Mylopoulos, & Pistore, 2004) and task (Omicini, 2000), and diagrams with notations for viewing them. The work is illustrated in a complete agent case study going through the phases of analysis and design.

The plan of the chapter is as follows. The next section describes the MESSAGE modelling language and process, followed by a section that describes the analysis case study and illustrates some of the graphical notation proposed. The high-level design case study is then presented. Next, we illustrate the organization-driven detailed design process. The final section presents an evaluation of MESSAGE based on analysis, design, and implementation experimentations.