Chapter X

Towards Radical Agent-Oriented Software Engineering Processes Based on AOR Modelling

Kuldar Taveter
The University of Melbourne, Australia

Gerd Wagner
Brandenburg University of Technology at Cottbus, Germany

Abstract

This chapter proposes a new agent-oriented software engineering process called RAP, which follows the Rational Unified Process (RUP) in many ways, but is based on Agent-Object-Relationship (AOR) modelling instead of object-oriented modelling. The chapter briefly presents the foundational ontology that supports the methodology and introduces the RAP/AOR viewpoint modelling framework. It then describes the modelling from the interaction, information, and behavior aspects of the framework by using a case study of business-to-business electronic commerce. Finally, the chapter describes an implementation approach based on the Model Driven
Architecture (MDA) and Extended Markup Language (XML). The methodology is aimed at the creation of distributed socio-technical systems consisting of both humans and technical, including software, components that may, in turn, include software agents.

Introduction

A Radical Agent-Oriented Process (RAP) defines a software engineering process using the Agent-Object-Relationship (AOR) modelling language proposed by Wagner (2003a). In AOR modelling, the agents in a problem domain are distinguished from the non-agentive objects. The agents’ actions, event perceptions, commitments, and claims, as well as their rights and duties, are explicitly included in the models.

The RAP/AOR methodology is based on Wagner (2003a) and Taveter (2004a). Wagner (2003a) presents an agent-oriented approach to the conceptual modelling of organizations and organizational information systems, called AOR modelling, where an entity is either an agent, an event, an action, a claim, a commitment, or an ordinary object, and where special relationships between agents and events, actions, claims, and commitments supplement the fundamental association, aggregation, and generalization relationship types of Entity-Relationship (ER) and UML class modelling. Business processes are viewed as social interaction processes emerging from the behaviour of the participating agents. In the proposed approach, behaviour is primarily modelled by means of interaction patterns expressed in the form of reaction rules that are visualized in interaction pattern diagrams.

Taveter (2004a) proposes an integrated business modelling technique—the Business Agents’ Approach—that is based on AOR modelling. Taveter (2004a) emphasizes that in addition to being a technological building block, an agent is an important modelling abstraction that can be used at different logical levels in the creation and development of an information system. The Business Agents’ Approach suggests an elaboration of the existing business modelling frameworks—six perspectives of agent-oriented business modelling for distributed domains. These perspectives are the organizational, informational, interactional, functional, motivational, and behavioural perspective. The Business Agents’ Approach covers modelling from all the perspectives mentioned by employing a combination of goal-based use cases, the AOR Modelling Language (AORML), and Object Constraint Language (OCL), forming a part of UML 2.0 (OMG, 2003b). The Business Agents’ Approach also extends the graphical notation of AORML by activity diagrams that are executable and enable to represent models of several or all perspectives in just one diagram.