Chapter 1.19
Combining Requirements Engineering and Agents

Angélica de Antonio
Universidad Politécnica de Madrid, Spain

Ricardo Imbert
Universidad Politécnica de Madrid, Spain

ABSTRACT

The concept of Agent is being used with different meanings and purposes in two separate fields of software engineering, namely Requirements Engineering and Agent-Oriented Software Engineering. After an introduction to Goal-Oriented Requirements Engineering (GORE) and its evolution into Agent-Oriented Requirements Engineering (AORE), this chapter provides a review of some of the main Agent-Oriented Software Engineering (AOSE) methodologies, focusing on their support for requirements modeling. Then the chapter analyzes how both approaches to Agents relate to each other, what the differences are among them, and how they could benefit from each other. Problems are identified and discussed that need to be addressed for a successful integration of both fields, and recommendations are provided to advance in this direction.

INTRODUCTION

This chapter is devoted to the analysis of a growing tendency to combine requirements engineering and agents. This analysis is conducted from a double perspective.

On one hand agents have been recognized as an abstraction that can be useful for requirements engineering (RE). Specifically, the concept of agent can be considered as a building block for structuring the description of an information system and the environment in which it will operate and with which it will interact. Agents are considered a nice abstraction since they can be used for modeling different kinds of entities, such as software, hardware, humans, or devices. From this point of view agents are a tool that can be used for engineering the requirements of any software system, be it agent-based or not. Agent-oriented requirements engineering (AORE) is considered as an evolution of goal-oriented requirements en-

gineering (GORE), both being social approaches to requirements engineering.

On the other hand agent-oriented systems, also known as multi-agent systems (MAS), are being increasingly recognized during the last few years (from the mid-'90s) as just the kind of software systems that need the application of software engineering practices for their development like any other software system, or even more if we take into account that MAS are complex systems and are usually applied to complex domains. That is how the term Agent-Oriented Software Engineering (AOSE) was coined a few years ago to describe a discipline that tries to define appropriate software engineering techniques and processes to be applied to these systems. The requirements of a MAS, like any other software system, need to be elicited, specified, analyzed, and managed, and the question that naturally arises is if engineering requirements for a MAS are different from any other software system.

Considering the apparent dissociation between the agent concept in GORE-AORE and in AOSE, we decided to investigate to which extent it would be possible to combine both approaches.

The second and third sections of this chapter describe the main approaches to the use of agents for requirements engineering, stating the principles underlying GORE and AORE. The fourth section analyzes how requirements engineering is currently being performed for agent-based systems. The last two sections show a reflection about the conclusions reached in our attempt to clarify how both approaches are related and how they could benefit one from the other.

GOAL-ORIENTED REQUIREMENTS ENGINEERING (GORE)

The initial requirements statements, which express customers' wishes about what the system should do, are often ambiguous, incomplete, inconsistent, and usually expressed informally. Many requirements languages and frameworks have been proposed for the refinement of the initial requirements statements, making them precise, complete, and consistent.

Increasingly information systems development occurs in the context of existing systems and established organizational processes. Some authors defend the need for an *early requirements analysis phase* with the aim to model and analyze stakeholder interests and how they might be addressed, or compromised, by various system-and-environment alternatives. This earlier phase of the requirements process can be just as important as that of refining initial requirements. However most existing requirements techniques are intended mainly for the later phase. Considerably less attention has been given to supporting the activities that *precede* the formulation of the initial requirements. These "early-phase" requirements engineering activities consider how the intended system would meet organizational goals, why the system is needed, what alternatives might exist, what the implications of the alternatives are for various stakeholders, and how the stakeholders' interests and concerns might be addressed. Early-phase RE activities have traditionally been done informally and without much tool support.

Because early-phase RE activities have objectives and presuppositions that are different from those of the late-phase, it seems appropriate to provide different modeling and reasoning support for the two phases.

The introduction of *goals* into the ontology of requirements models represented a significant shift toward this direction. Previously the world to be modeled consisted just of entities and activities. Goal analysis techniques have proved to be very useful, covering functional and non-functional goal analysis.

Some of the most remarkable GORE approaches are EKD (Enterprise Knowledge Development) (Kavakli & Loucopoulos, 1998) and KAOS (Dardenne, van Lamsweerde, & Fickas, 1993; van Lamsweerde, Darimont, & Letier, 1998;