Chapter XXI
Using Rule-Based Concepts as Foundation for Higher-Level Agent Architectures

Lars Braubach
University of Hamburg, Germany

Alexander Pokahr
University of Hamburg, Germany

Adrian Paschke
Freie Universität Berlin, Germany

ABSTRACT

Declarative programming using rules has advantages in certain application domains and has been successfully applied in many real world software projects. Besides building rule-based applications, rule concepts also provide a proven basis for the development of higher-level architectures, which enrich the existing production rule metaphor with further abstractions. One especially interesting application domain for this technology is the behavior specification of autonomous software agents, because rule bases help fulfilling key characteristics of agents such as reactivity and proactivity. This chapter details which motivations promote the usage of rule bases for agent behavior control and what kinds of approaches exist. Concretely, these approaches are in the context of four existing agent architectures (pure rule-based, AOP, Soar, BDI) and their implementations (Rule Responder, Agent-0 and successors, Soar, and Jadex). In particular, this chapter emphasizes in which respect these agent architectures make use of rules and with what mechanisms they extend the base functionality. Finally, the approaches are generalized by summarizing their core assumptions and extension mechanisms and possible further application domains besides agent architectures are presented.

INTRODUCTION AND MOTIVATION

Software agents are computational entities that can act autonomously without user intervention and interact with each other in order to fulfill given tasks. In the recent years, agent technology has evolved in to a large and highly active research field. Therefore, nowadays many different forms of agent applications exist, such as information agents, interface agents, mobile agents, and agents for problem solving (Nwana 1995), many of which also have been put successfully into practice (Jennings and Wooldridge 1998). Besides the application domain, implemented agent systems can also be distinguished by the employed 'agent architecture', i.e. the control structures that facilitate the specification and execution of agent behavior. Often these agent architectures build upon or have been influenced by rule-based technology.

In this chapter, rule-based technology is seen as a declarative approach to programming. On a conceptual level, a rule-based approach enforces a separation of the state of a system (i.e. the working memory) from the behavior (i.e. state transition rules). Different types of rules exist for implementing different types of systems. E.g. in production systems so called forward chaining rules are used, which are composed of a condition and an action part, such that the system will evaluate the condition of a rule and execute the corresponding action, when the condition holds. On the other hand, backward chaining rules (containing antecedence and consequence parts) are used in expert systems and allow deriving new knowledge (consequence) from existing facts (antecedence). For implementing such systems, sophisticated mechanisms have been developed for e.g. representing knowledge in the state, efficiently evaluating rules, deciding which rule to execute when multiple rules match at the same time (conflict resolution) and dealing with knowledge created by rules, which are no longer activated (truth maintenance).

The general approach of this chapter is to present and discuss agent architectures as one very interesting application domain for rule concepts. Therefore, the chapter shows the current state of the art with respect to rules for describing behavior of intelligent agents. One specific objective of this chapter is to show why rule concepts are important for the specification of agent architectures and in what different ways they can be used as basis for such architectures. On a more generic level, the objective of this chapter is to explain how higher-level concepts can be built on rule concepts and what advantages can be drawn from such developments.

The outline of the chapter is as follows. In the next section a short overview of the field of agents and multi-agent systems will be given, with a special focus on the role of agent architectures. Section 3 will provide a description scheme for and a categorization of existing agent architectures with regard to their incorporation of rule-based technology. An in-depth discussion of each of the four categories, including a detailed analysis of one representative in each case, follows in Sections 4-7. Section 8 will provide an outlook on interesting areas of future research before Section 9 closes the chapter with a summary and conclusion.

BACKGROUND ON AGENTS AND MULTI-AGENT SYSTEMS

The field of agent technology emerged during the Nineties of the last century and has its roots in different areas of computer science such as artificial intelligence (AI), software engineering (SE), and distributed computing (Luck et al. 2005). In agent technology, an agent is seen as an independent software entity situated in an environment that is capable of controlling its own behavior (i.e. an agent can act without user intervention). Although agent technology is a very diverse field with many sometimes quite unrelated sub areas, general consensus exists, that agents can be ascribed the following set of properties (Wooldridge 2001):

- *Autonomy.* An agent decides on its own, how to accomplish given tasks. This is also the case for agents that act on behalf of a user.
- *Reactivity.* An agent continually monitors its environment and automatically reacts to changes in a timely manner, if necessary.