ABSTRACT

Multi-agent systems are viewed as consisting of individual agents whose behaviors are regulated by an organizational artifact. This chapter presents a programming language that aims at facilitating the implementation of norm-based organizational artifacts. The programming language is presented in two steps. We first present a programming language that is designed to support the implementation of non-normative organizational artifacts. These artifacts are specified in terms of non-normative concepts such as the identity of participating agents, the identity of the constituting environments in which individual agents can perform actions, and the agents’ access relation to the environments. The programming language is then modified and extended to support the implementation of norm-based artifacts. Such artifacts are specified in terms of norms being enforced by monitoring, regimenting, and sanctioning mechanisms. The syntax and operational semantics of the programming language are discussed and explained by means of a conference management system example.
INTRODUCTION

In this chapter, multi-agent systems are considered as consisting of individual agents that are autonomous and heterogeneous. The first assumption implies that each individual agent pursues its own design objectives and the second one implies that the internal states and operations of individual agents may not be known to external entities (Zambonelli, Jennings, & Wooldridge, 2003; Esteva, Rodríguez-Aguilar, Rosell, & Arcos, 2004). In order to achieve the overall goal of such multi-agent systems, the observable/external behavior of individual agents and their interactions should be regulated.

There are two main approaches to regulate the external behavior of individual agents. The first approach is based on coordination artifacts that are specified in terms of low-level coordination concepts such as synchronization of processes (Ricci, Viroli, & Omicini, 2007). The second approach is motivated by organizational models, normative systems, and electronic institutions (Searle, 1995; Jones & Sergot, 1993; Esteva et al., 2004; Grossi, 2007). In this approach, norms are used to regulate the behavior of individual agents.

Using a social and normative perspective is conceived as a way to make the development and maintenance of multi-agent systems easier to manage. A plethora of social concepts (e.g., roles, social structures, organizations, institutions, norms) has been introduced in multi-agent system methodologies, such as Gaia (Zambonelli et al., 2003), models such as OperA (Dignum, 2004), Moise+ (Hübner, Sichman, & Boissier, 2002), and electronic institutions and frameworks such as AMELI (Esteva et al., 2004) and S-Moise+ (Hübner et al., 2002). See also Chapters VI, VIII and XIV for other models that specify multi-agent systems in terms of social and organizational concepts.

Norms in multi-agent systems can be used to specify the standards of behavior that agents ought to follow in order for the overall objectives of the system to be met. However, to develop a multi-agent system does not boil down to state a number of standards of behavior in the form of a set of norms, but rather to organize the system in such a way that those standards of behavior are actually followed by the agents. This can be achieved by regimentation or enforcement mechanisms (Jones & Sergot, 1993).

When regimenting norms all agents’ external actions leading to a violation of those norms are made impossible. Via regimentation (e.g., gates) the system prevents an agent from performing a forbidden action (e.g., entering an underground train platform without a ticket). However, regimentation drastically decreases the agent’s autonomy. Instead, enforcement is based on the idea of responding after a violation of the norms has occurred. Such a response, which includes sanctions, aims to return the system to an acceptable state. Crucial for enforcement is that the actions that violate norms are observable by the system (e.g., fines can be issued only if the system can detect travelers entering the underground system without a ticket).

The main contribution of this chapter is to present and discuss a simplified version of a programming language that is designed to implement norm-based artifacts that can be used to regulate the behaviors of individual agents constituting a multi-agent system. Such artifacts are assumed to be used by individual agents to perform their external actions in the shared environment(s) and to pass messages to each other. In the next section, we present the syntax and semantics of a programming language that can be used to specify a (non-normative) multi-agent system. This programming language is then modified and extended to allow the implementation of norm-based artifacts. The modified programming language provides constructs to represent norms and mechanisms to enforce them. Then, an example is given to illustrate the use of the programming language. The chapter will be closed by discussing some norm-based approaches to multi-agent systems available at the moment in the literature, among