Engineering Applications of Artificial Intelligence 26 (2013) 76–95

Contents lists available at SciVerse ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

MaNEA: A distributed architecture for enforcing norms in open MAS

N. Criado [a,*], E. Argente [a], P. Noriega [b], V. Botti [a]

[a] Departamento de Sistemas Informáticos y Computación, Universitat Politècnica de València, Camino de Vera s/n. 46022, Valencia, Spain
[b] Institut d'Investigació en Intel.ligència Artificial, Consejo Superior de Investigaciones Científicas, Campus de la UAB, Bellaterra, Catalonia, Spain

ARTICLE INFO

Article history:
Received 2 December 2011
Received in revised form
17 July 2012
Accepted 21 August 2012
Available online 30 September 2012

Keywords:
Norm enforcement
Agent platforms
Open MAS

ABSTRACT

Norms have been promoted as a coordination mechanism for controlling agent behaviours in open MAS. Thus, agent platforms must provide normative support, allowing both norm-aware and non-norm-aware agents to take part in MAS that are controlled by norms. In this paper, the most relevant proposals on the definition of norm enforcement mechanisms are analyzed. These proposals present several drawbacks that make them unsuitable for open MAS. In response to these problems, this paper describes a new Norm-Enforcing Architecture aimed at controlling norms in open MAS. Specifically, this architecture supports the creation and deletion of norms on-line as well as the dynamic activation and expiration of instances. Finally, it can dynamically adapt to different scale MAS. The efficiency of this architecture has been experimentally evaluated and the results are shown in this paper.

1. Introduction

The main applications of Multi-Agent Systems (MAS) support large scale open distributed systems. These systems are characterized by the heterogeneity of their participants; their limited trust; a high uncertainty; and the existence of individual goals that might be in conflict (Artikis and Pitt, 2001). In these scenarios, norms are conceived as an effective mechanism for achieving coordination and ensuring social order; i.e., norms represent an effective tool for regulating the actions of software agents and the interactions among them (López et al., 2006). Most of the proposals on methodologies and guidelines aimed at developing open MAS (Argente et al., 2011b; Dignum et al., 2005) are based on organizational concepts, such as norms. These concepts facilitate the analysis and design of coordination and collaboration mechanisms for MAS. Therefore, norms should be considered in the design and specification of the MAS (Criado et al., 2011b). As pointed out in Castelfranchi (2003), the use of norms in MAS allows better results to be achieved in dynamic and complex environments. Specifically, the fact that agents can violate norms autonomously allows a better adaptation to the environmental changes. Finally, the occurrence of norm violations can evidence the need to adapt the MAS (Bernon et al., 2003). Agent platforms are the software that supports the development and execution of MAS. Thus, norms must be also considered in the design and implementation of agent platforms (Criado et al., 2011b). As a consequence, agent platforms must implement norms in an optimized way, given that in open MAS the internal states of agents are not accessible (Criado et al., 2011a). Therefore, norms cannot be imposed as agent's beliefs or goals, but they must be implemented in the platforms by means of *control* mechanisms (Grossi et al., 2007).

This paper shows an overview of the most relevant works on norm implementation. This paper considers the main challenges of open MAS and points out the main deficiencies and drawbacks of agent platforms and infrastructures when supporting norms. With the aim of overcoming some of these problems, in this paper a Norm-Enforcing Architecture, known as MaNEA, is proposed. Specifically, MaNEA has been integrated into the Magentix2 platform.[1] The Magentix2 platform allows the management of open MAS in a secure and optimized way. Its main objective is to bring agent technology to real domains: business, industry, e-commerce, among others. This goal entails the development of more robust and efficient mechanisms for enforcing norms that control these complex applications.

This paper is organized as follows: Section 2 contains the analysis of the main proposals on norm enforcement; Section 3 describes briefly the Magentix2 platform; Section 4 describes the main components of MaNEA; Section 5 illustrates the performance of MaNEA through a case study; Section 6 contains an evaluation of this architecture; and, finally, Section 7 contains conclusions and future works.

2. Related work

Most of the proposals on norms for controlling MAS tackle this issue from a theoretical perspective (Boella, 2004; Sergot, 1998). However, there are also works on norms from a computational point of view. These works propose control mechanisms for norms to have

* Corresponding author.
E-mail addresses: ncriado@dsic.upv.es (N. Criado), eargente@dsic.upv.es (E. Argente), pablo@iiia.csic.es (P. Noriega), vbotti@dsic.upv.es (V. Botti).

0952-1976/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.engappai.2012.08.007

[1] http://magentix2.gti-ia.upv.es/.

an effective influence on agent behaviours (Grossi et al., 2007). These *control* mechanisms are classified into two categories (Grossi et al., 2007): *regimentation* mechanisms, which consist of making the violation of norms impossible; and *enforcement* mechanisms, which are applied after the detection of norm violations, reacting upon them.

2.1. Norm regimentation

Regimentation forces ideality (expressed as norms) and reality (defined by agents' behaviour) to coincide (Jones and Sergot, 1993). Thus, regimentation mechanisms prevent agents from performing forbidden actions by mediating the resources and the communication channel, such as Electronic Institutions (EIs) (Esteva et al., 2004). However, the regimentation of all actions can be not only difficult or impossible, but also it is sometimes preferable to allow agents to violate norms (Castelfranchi, 2003). In fact, Open MAS that operate in complex and changing environments may benefit from the occurrence and detection of norm violations. The reasons behind desirability of norm violations are that it is impossible to take a thorough control of all agents' actions; or agents could obtain higher personal benefits when norms are violated; or norms may be violated by functional or cooperative motivations. For example, changes in the environment may cause norms to lose their validity. In this situation, autonomous agents may deviate from norms in order to achieve their own goal, which may imply a better performance of the whole MAS. All these situations require norms to be controlled but allowing norm violations to occur. In response to this need, the *enforcement* mechanisms have been developed, which are explained in the following subsection.

2.2. Norm enforcement

Proposals on the enforcement of norms can be classified according to the entity that observes whether norms are fulfilled or not. Specifically, norm compliance must be observed by agents or the infrastructure may provide mechanisms for monitoring agent activities according to norms. Each one of these approaches to norm enforcement is explained below.

2.2.1. Agent observability

This approach is characterized by the fact that norm violations may be observed by agents that are involved in an interaction in which the norm has been violated (*second-party* observability), or other agents that observe an interaction in which they are not directly involved (*third-party* observability).

Second-party observability

There are proposals (such as Venkatraman and Singh, 1999; Daskalopulu et al., 2002) in which the agents involved in an interaction are responsible for monitoring norms. In these approaches, agents evaluate their interaction partners subjectively. In accordance with this evaluation, agents may punish or reward their partners (Boella, 2003) or they may start a grievance procedure (Criado et al., 2010b). For example, Venkatraman and Singh (1999) propose an approach for testing the compliance of agents with respect to a commitment. Commitments are specified in temporal logic and their compliance is evaluated with respect to locally constructed models for the given observer. The work contained in Daskalopulu et al. (2002) proposes a framework for contract performance arbitrating. In particular, it uses subjective logic (Jøsang, 2001) as the formal basis for evidence-based reasoning. Subjective logic addresses the problem of forming a measurable belief about a proposition on the basis of insufficient evidence, or in the presence of uncertainty and ignorance.

Third-party observability

If there are agents that are not directly involved in an interaction but that are capable of observing it, they would be also capable of forming an own *image* about the interacting participants. Moreover, these evaluations or *reputations* may be exchanged. Thus, agents are persuaded to obey norms because their non-normative behaviour can be observed by others. In this case, the *society* as a whole acts as norm enforcer (Sen and Airiau, 2007). These non-compliant agents might even be excluded from the society (de Pinninck et al., 2007). The role of emotions in social enforcement (Elster, 1996) is also interesting. For example, the work described in Fix (2006) models the emotion-based enforcement of norms in agent societies. In this approach, the whole society observes compliance of norms and generates social emotions such as contempt or disgust, in case of norm violation; and admiration or gratefulness, in case of norm fulfilment. In the same way, agents observe the expression of these emotions and are also able to generate emotions such as shame or satisfaction in response.

The main drawback of proposals on second-party and third-party observability is the fact that the underlying infrastructure does not offer support for enforcing norms. Thus, the norm monitoring and the reaction to violations must be implemented by agent programmers. In this sense, agent programmers are responsible for watching over norm compliance. Even if the infrastructure provides authority entities that act as *arbiters* or *judges* in grievance processes, agents must be endowed with capabilities for both detecting norm violations and participating in these dispute resolution processes.

2.2.2. Infrastructural observability

Normative agent platforms provide entities that are in charge of both observing and enforcing norms. Cardoso and Oliveira (2007) propose a norm-enforcing architecture in which the monitoring and enforcement of norms is made by a single institutional entity, named as *normative environment*. This entity receives all messages that have been exchanged among agents and determines if an agent has violated (vs. fulfilled) a norm. In this case, the *normative environment* sends a sanctioning (vs. rewarding) notification to this agent. As argued by Cardoso and Oliveira the implementation of the *normative environment* as a centralized component represents a performance limitation when dealing with a considerable number of agents.

To address the performance limitation of centralized approaches, distributed mechanisms for an institutional enforcement of norms are proposed in Minsky and Ungureanu (2000) and Gaertner et al. (2007). These works propose languages for expressing norms and software architectures for the distributed enforcement of these norms. Minsky and Ungureanu (2000) present an enforcement mechanism that is implemented by the Moses toolkit (Minsky and Ungureanu, 1998). Its performance is as general (i.e., it can implement all norms that are controllable by a centralized enforcement) and more scalable and efficient than centralized approaches. However, one of the main drawbacks of this proposal is the fact that norms can only be expressed in terms of the messages sent or received by an agent; i.e., this framework does not support the definition of norms that affect an agent as a consequence of an action carried out independently by another agent. This problem is overcome by Gaertner et al. (2007). In their approach, Gaertner et al. propose a distributed architecture for enforcing norms in EI. Specifically, this architecture only controls dialogical actions. Thus, the dialogical actions performed by agents cause the propagation of normative propositions (i.e., obligations, permissions, and prohibitions). These normative propositions are taken into account by the normative level; i.e., a higher level in which norm reasoning and management processes are performed in a distributed manner.