A Framework for Supporting Simulation with Normative Agents

Marx Viana[1], Francisco Cunha[1], Baldoino Santos Neto[2], Paulo Alencar[3], Carlos J. P. de Lucena[1]

[1] Pontifical Catholic University (PUC-Rio) - Rio de Janeiro – RJ, Brasil

[2]Federal University of Alagoas - Alagoas – AL, Brasil

[3]University of Waterloo - Waterloo - Ontario, Canada

`{mleles, fplacido, lucena}@inf.puc-rio.br, baldoino@ic.ufal.br, palencar@uwaterloo.ca`

Abstract. *Multi-agent systems are societies in which autonomous agents work to achieve both common and individual goals. In this context, norms have been used as mechanisms to regulate the behavior of such agents and ensure a desirable social order where agents can work together. Although norms are a promising mechanism, it is necessary to evaluate the positive and negative impact of using them. Therefore, this work presents a framework for normative multi-agent systems simulation that provides the mechanisms necessary to understand the impact of norms on agents and the society they belong to.*

1. Introduction

Multi-agent systems are societies where entities, known as agents, are autonomous, heterogeneous and can work to achieve common or disparate goals [Wooldridge 2011]. In order to deal with autonomy and diversity of interests among the different members, such systems provide a set of norms that is used as a social control mechanism to ensure that a desirable social order in which agents can work is maintained [Silva 2008]. Although norms are promising mechanisms to regulate the behavior of agents, one should take into account that they are autonomous and, therefore, free to decide to fulfill or violate each system norm. This type of agent reasoning refers to normative strategies.

However, in order to discourage violations of specific norms by the agents, designers can provide rewards when the agents fulfill these norms and punishments when the norms are violated [Silva 2008]. In this context, there is a need for mechanisms that enable designers to develop and verify the impact of norms (i.e., rewards and punishments) as well as normative strategies not only on individual agents, but also on the whole society.

This paper proposes a framework (Normative Agent Java Simulation Framework, which we call JSAN) that supports the evaluation of norms through normative multi-agent system simulations. The JSAN framework provides the necessary mechanisms to understand the impact of norms on agents that adopt some given strategies to deal with these norms. These mechanisms enable (i) implementing of different types of normative strategies; (ii) collecting information about the simulated environment; (iii) displaying information about the simulated scenarios; (iv) supporting the implementation of different movement strategies for the agents in the environment; (v) generating norms; and (vi) generating agent goals. This framework extends of JASON [Bordini et al. 2007], which enables the development and implementation of BDI agents (Belief, Desire, and Intention) [Machado and Bordini 2002]. We plan to evaluate the applicability of the JSAN Framework in two scenarios: the first is related to the mission of rescuing civilians who are in geoenvironmental risk areas [Cerqueira et al. 2009] and the second involves crime prevention [Bosse and Gerritsen 2010].

2. Motivation

The creation of JSAN Framework was due to the need to develop simulations in which it would be possible to insert norms to restrict the behavior of agents in a given environment. In addition, JSAN provides the necessary mechanisms for agents to be able to understand and deal with these norms. In the next paragraphs two scenarios are described to illustrate the application of JSAN to assist in decision-making, showing possible cases where norms are inserted to restrict the behavior of agents in the environment.

In a first application scenario, the simulation framework can be used to support the rescue of civilians who are in geoenvironmental risk areas. It is known that landslides are natural phenomena that are often difficult to predict since they depend on many factors (e.g., slope angle, climate, water content, vegetation) and complex relationships between these factors. The annual number of landslides is in the thousands in the city of Rio de Janeiro, and the infrastructure damage is worth more than a billion dollars [Santos Neto and Lucena 2010]. In this setting, there is a need to build platforms to assist experts in two areas, namely the analysis of geoenvironmental risk areas and the evacuation planning of civilians located in these areas [Cerqueira et al. 2009].

The evacuation planning can be assisted by simulations using the JSAN Framework, which aims to implement scenarios involving the creation of situations where civilians are in geoenvironmental risk areas and, thus, to provide different strategies for firefighter agents, which are regulated by norms, to rescue these civilians. The simulations are normative multi-agent systems that receive data about the geoenvironmental risk areas, such as weather conditions, information about the existence of civilians in geoenvironmental risk areas, redemption forms for the withdrawal of civilians from these risky locations (with troops, land vehicles or aircraft), norms that firefighter agents must follow during the rescue operation, and rescue plans to be used in the simulation. The supplied data is received by the Manager Agent (Chief Fireman) responsible for sending this information to the firefighter agents who are able to find different solutions to evacuate civilians from geoenvironmental risk areas.

In a second application scenario, the simulation framework can be used in the context of crime prevention, in which an important challenge is the analysis of the displacement of crime. Some of the many problems in this area relate to the prediction and prevention with respect to areas with a potentially high crime rate, making the study of the displacement of crime a promising research area [Bosse and Gerritsen 2010]. Certain types of crimes typically happen around specific locations in a city, especially in streets nearby shopping malls, train stations and highways. These areas are known as sites with high crime rates. However, these sites do not stay the same for a long period.

A number of known factors can cause the displacement of these criminality sites to another region. For example, the deployment of cameras in train stations can cause crime rates to decrease in that area [Bosse and Gerritsen 2010]. The insertion of norms in crime prevention support aims to regulate the actions of the police agents [Bosse and Gerritsen 2010]. For example, such norms may influence them to act to prevent the largest possible number of robberies, but not to think about their safety. Specifically, if a norm governing the police agent involves the case when he or she wishes to go to an area with many criminals and a few police agents, this norm could describe that it is necessary that the agent making the arrest calls for backup. In this way, if the agent complies with the norm, others police agents will go to the location and the arrest will be carried out successfully, and the agents will receive a reward that allows them to travel at a higher speed on the move to their targets. However, if the police agent violates the norm and puts his or her life at risk, their travel speed is reduced and they will lose some of their ability to make arrests.

3. Norms Definition

In this article, we adopted the definition of norms presented in [Lopez 2003] as: *Norm* (*addressees, deonticConcept, activation, deactivation, reward, punishments, elementRegulated*), where *addresses* refer to the set of agents that will be governed by the norms and *deonticConcept* is the deontic concept associated with the norm. *Activation* and *deactivation* are the activation and deactivation of the norm in the environment respectively. *Rewards* and *punishments* are the rewards and punishments attached to norm in the case it is fulfilled or violated, respectively. Finally, the entity governed by the norm is defined by the *elementRegulated* attribute.

To understand the definition of norms better, imagine that a firefighter receives the mission of rescuing civilians who are in geoenvironmental risk areas, and attempts to perform this rescue in accordance with the norms that were addressed. At this time the following norm is sent to the firefighter agents, "protect the lives of civilians at geoenvironmental risk areas" with the following attributes: the addressees are the firefighter agents, the required deontic concept is obligation, their reward when a norm is met is that the agent will get air or ground support in his mission, the punishment in case the norm is violated is that the firefighter agents will not get support in their rescue operation, the norm is activated if there is any person at risk, the norm is deactivated when all civilians are safe, and the element regulated by the norm is the action of using aircrafts because of the rescue costs.

4. Related Work

The n-BDI architecture [Criado et al. 2010] presents a model for designing agents capable of operating in environments governed by norms. This architecture considers that the selection of objectives should be performed based on the priority associated with each objective, where this priority is determined taking into account the priority of the norms that govern the objective. However, it is not clear in this approach how the components of a norm can be evaluated. In addition, the approach does not support a strategy to deal with the conflicts between norms.

In [Lopez and Marquez 2004], the authors propose a formal model, using the Z formal specification language, for modeling agents able to achieve their objectives taking into account the norms of the system. According to [Lopez and Marquez 2004] an agent created from such a model is able to: (i) check if it is the one responsible for fulfilling a norm; (ii) verify the activation and deactivation of a norm taking into account the beliefs of the agent; (iii) evaluate and decide to fulfill or violate every norm of the system; and (iv) make the decision to fulfill or violate a norm, removing or adding agent goals. Besides not showing how the evaluation of a norm is performed, the authors do not focus on identifying and resolving conflicts between norms, checking fulfilled or violated norms, and showing the influence of norms on the plan selection process and intentions of the agents.

In [Lopez 2002], the author presents a set of strategies that can be adopted by agents to deal with norms. These strategies are: *Social*, *Pressured*, *Opportunistic*, and *Rebellious*. The *Social* strategy focuses on the agents complying with the rules without worrying about their individual goals. The *Pressured* strategy happens when agents fulfill the norms considering only the punishments will harm them to achieve their individual goals. Another strategy is the *Opportunistic* strategy, in which agents consider only the effects of rewards on their individual goals, and seek to fulfill only the norms for which the rewards of the individual goals are more important than those of the social goals. The *Rebellious* strategy implies that the agents will care only to achieve their individual goals, regardless of the punishments attached to the violation of the norms. Finally, the *Selfish* strategy is the combination of the *Pressured* and the *Opportunistic* strategies.

5. Normative Agent Java Simulation Framework - JSAN

JSAN provides the ability to create simulations that help to understand the impact of norms on agents capable of adopting different strategies to deal with the agent-related norms. Therefore, the framework enables the implementation of different normative strategies, collecting information of the simulated environment, and displaying information about the simulated scenarios that support the implementation of different movement strategies of the agents in the environment and supporting the generation of the norms and goals of the agents. The class diagram presented in Figure 1 shows the main classes and methods of the framework. Agents are represented by the class *NormativeAgent*, which is an extension of the original *Agent* class in the JASON framework [Bordini et al. 2007]. In addition, the simulation environment is described by the *EnvironmentSimulation* class, which extends the *Environment* class and provides support to create the simulation environment.

The *ExecuteAction* method is an extension of *Environment*, and much of the *Environment* code is written in it. Whenever an agent tries to perform an essential action, its identification and their chosen actions are passed to this method. For this reason, the code *ExecuteAction* method must verify that the action is valid and then do what is necessary for the action to be performed. The action may change the perceptions of agents. If this method returns true it means that the action was performed successfully. The *EnvironmentSimulation* class is also responsible for managing the creation of norms and individual goals of the agents, using instances of *GenerateNormsStrategy* and *GenerateGoalStrategy* classes, respectively. In addition, *EnvironmentSimulation* is an extension of the *MovementStrategy* class that is responsible for managing the strategies used by the agents to move in the environment as events occur during the simulation. Each norm in environment is an extension of the

Norm class (see the definition of norms provided in Section 3). Additionally, the framework provides a mechanism to report the impact of the norms on normative agents. To use this mechanism is necessary to extend the *ReportStrategy* class, passing the following parameters: (i) the environment in which the simulation is being carried out and (ii) an implementation of *NormStrategy* class, which contains the strategy used by the agent to handle the norms.

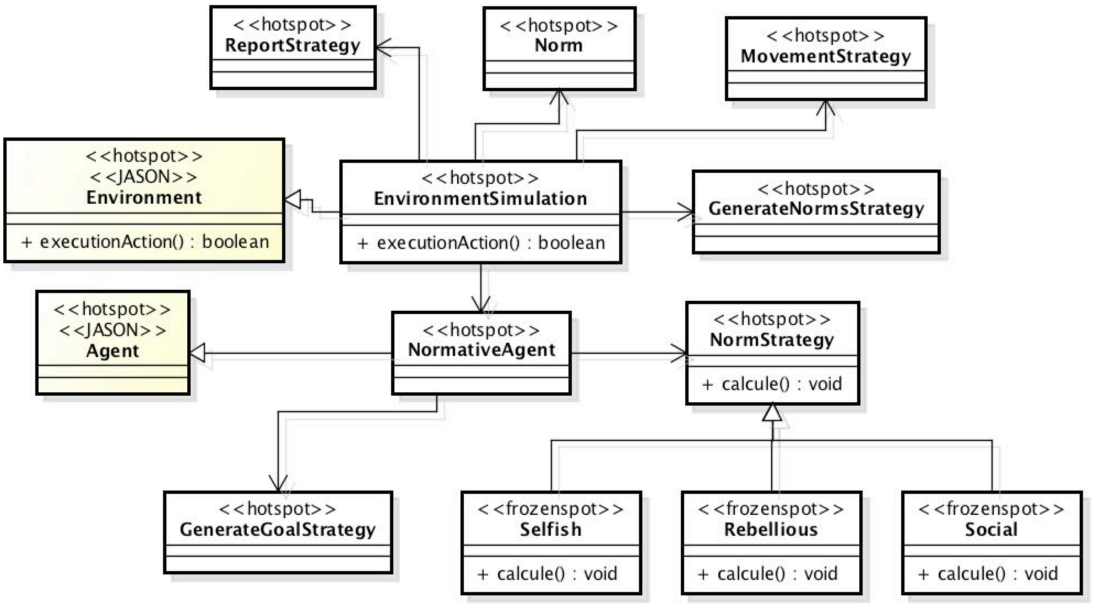

Figure 1 - Class Diagram

In short, to build a simulation of normative multi-agent systems using this framework it was necessary to extend some other classes. The *GenerateNormsStrategy* class needed to be extended to generate the norms that will be activated in the simulation environment. Further, the *GenerateGoalStrategy* class was extended so that we can generate the goals of the agents. By extending the *MovementStrategy* class, it is possible to generate different movement strategies for the agents in the simulated environment. In the *NormStrategy* class, a method called *calculate* was defined, which needed to be implemented to describe the strategy that will be used for agents dealing with the norms. We implement three types of strategies to deal with norms (*Selfish*, *Rebellious* and *Social*) (See definitions in Section 4). The *ReportStrategy* class was extended to report accurate information about each simulation.

7. CONCLUSION

This paper proposes a framework that allows the construction of simulations involving normative agents and provides the necessary mechanisms to understand the impact of norms on agents that adopt specific strategies to deal with these norms. Additionally, the framework provides many ways to represent computational normative concepts that can be used for better understanding norms related to the behavior and regulation of the agents. The proposed structure makes explicit what role the norm has in a society and the elements regulated by it, which in turn can be used by agents for decision-making in the society in which they live.

In order to evaluate the applicability of the framework, it will be applied in the evacuation of civilians from geoenvironmental risk areas [Cerqueira et al. 2009], and in crime prevention [Bosse and Gerritsen 2010]. For example, the insertion of norms in the simulation environment, which implements this framework, can be used to regulate the behavior of firemen agents and make them able to be norm-aware in the process of rescuing civilians from geoenvironmental risk areas.

As future work we plan to implement new mechanisms to deal with different levels of agent autonomy in order to show how different restriction levels and communities can influence the satisfaction of a norm application [Lopez 2002]. In the current version of the framework the autonomy-related restriction levels was not taken into account. However, the framework can be enhanced with different levels of restrictions, thus offering the possibility to achieve better results in terms of promoting a desirable social order.

References

Bordini, R. H., Hübner, J. F., & Wooldridge, M. (2007) "Programming multi-agent systems in AgentSpeak using Jason" (Vol. 8). John Wiley & Sons.

Bosse, T.; Gerritsen, C. (2010) "An Agent-Based framework to Support Crime Prevention", AAMAS, Toronto. 525-532.

Cerqueira, S. L. R., Santos Neto, B. F., Lucena, C. J. P., Campos, T. M. P., Moncada M. P. H. (2009) "*Plataforma GeoRisc Engenharia da Computação Aplicada à Análise de Riscos Geo-ambientais*". PUC-RIO. Rio de Janeiro. Brasil.

Criado, N., Argente, E., Noriega, P., and Botti, V. (2010) "Towards a Normative BDI Architecture for Norm Compliance". COIN@ MAL- LOW2010.

Lopez, F. L.; Luck, M.; D'Inverno, M. (2002) "Constraining Autonomy through Norms", AAMAS.

Lopez, F. L. (2003) "Social Power and Norms" (Doctoral dissertation, University of Southampton).

Lopez, L. F. Marquez, A. A. (2004) "An Architeture for Autonomous Normative Agents", IEEE, Puebla, México.

Machado, R. Bordini, R. H. (2002) "Running AgentSpeak(L) agents on SIM AGENT", (ATAL-2001), August 1–3.

Santos Neto, B. F. D. Lucena, C. J. P. (2010) "JAFF: implementando agentes auto-adaptativos orientados a serviços", (PUC-Rio) Rio de Janeiro - Brasil.

Silva, V. T. (2008) "From the specification to the implementation of norms: an automatic approach to generate rules from norms to govern the behavior of agents". Autonomous Agents and Multi-Agent Systems, 17(1), 113-155.

Wooldridge, M. (2011). "Introduction to Multiagent Systems". John Wiley & Sons, Inc., New York, NY, USA. 1, 2.1, 2.7, 4.1, 4.1.1, 6.1.