A Modeling Environment for Normative Multi-Agent Systems

Emmanuel S. S. Freire, Robert M. Rocha Júnior and Mariela I. Cortés
GESSI – Grupo de Engenharia de Software e Sistemas Inteligentes, Computer Science Department, Universidade Estadual do Ceará (UECE), Fortaleza, Ceará, Brazil

Keywords: Normative Multi-Agent Systems, Norms, NorMAS-ML, MAS-ML Tool, CASE.

Abstract: Tools to support the development of Multi-Agent Systems are essential to promote the effective utilization of the approach in the industry. More specifically, an adequate support for the modeling activity is useful to increase the productivity and contributes to the reduction of possible errors that would only be detected in execution time. This paper presents the evolution of MAS-ML tool to support the modeling of normative multi-agent systems modeled in NorMAS-ML language.

1 INTRODUCTION

Like others paradigms, agent-oriented development requires adequate techniques for exploring its benefits and properties in the construction and maintenance of such software (Zambonelli et al., 2001). In this scenario, tools to support the activities of engineering (CASE) (Sommerville, 2007) become decisive factors for the choice and effective use of available methods and techniques. In particular, modeling tools contribute to increase the productivity of developers and ensuring good formation models for construction.

In Multi-Agent System (MAS), agents are inserted in environments where they interact with others agents to achieve their goals. In complex scenarios, the agent's behavior is governed by norms, which aims to govern the agent actions through the concepts of obligation, permission and prohibition (Figueiredo and Silva, 2010).

Norms can be defined both modeling time as at runtime. However, the definition in modeling time carries a deeper and more comprehensive understanding of the structure, as well as an overview that allows observing the behavior of the system, before being executed. Therefore, the use of a language that allows the modeling of norms along with entities in MAS is fundamental.

The modeling language NorMAS-ML (Normative Multi-Agent System Modeling Language) (Freire et al., 2012) represents an extension of MAS-ML (Silva et al., 2007) that brings the necessary mechanisms to support the modeling of the main entities found in MAS along with the static properties of norms. MAS-ML has a modeling environment, MAS-ML tool (Feijó, 2012), that allows modeling all static diagrams and the dynamic diagram of sequence originally defined in the language. However, in the current version, the tool does not support the norm concepts introduced in NorMAS-ML, as well the static diagram for modeling norms and their properties.

This paper aims to present the evolution of MAS-ML tool in order to model the norm diagram. The paper is structured as follows: Section 2 presents related works. The language NorMAS-ML and the modeling environment MAS-ML tool are described in Section 3. Section 4 discusses the evolution of MAS-ML tool to enable the modeling the norm diagram. A case study illustrating the use of the tool is presented in Section 5. Finally, conclusions and future works are discussed in Section 6.

2 RELATED WORK

In this section, we analyze the modeling tools candidate to extension for supporting NorMAS-ML, i.e., incorporating the adequate support for modeling of norms along with the entities of the MAS.

VisualAgent (De Maria et al., 2005) is a software development environment that aims support the developers in the specification, project and development of MAS. The tool is based in the MAS-ML metamodel (Silva et al., 2007) and proposes a model driven approach for development of MAS

S. S. Freire E., M. Rocha Júnior R. and I. Cortés M..
A Modeling Environment for Normative Multi-Agent Systems.
DOI: 10.5220/0004438604510458
In *Proceedings of the 15th International Conference on Enterprise Information Systems* (ICEIS-2013), pages 451-458
ISBN: 978-989-8565-59-4

using the ASF (Agent Society Framework) (Silva, Cortés and Lucena, 2004), that represents the main point of the approach. However, the tool does not enable the checking of the models in relation to MAS-ML metamodel, neither the modeling of the normative concepts.

NormML Tool Kit (Figueiredo, 2011) is a modeling environment based in the NormML language that supports the mechanisms for model checking and checking for conflicts between norms. The environment is composed for two plugins for Eclipse framework (Eclipse Platform, www.eclipse.org): NormML Editor and NormML Conflict Checker. However, only the modeling of the normative elements is supported, thus the modeling of the typical entities of MAS and their properties is not foreseen.

Finally, MAS-ML tool (Feijó, 2012) is a modeling environment that allows the modeling of MAS according to the specification of the MAS-ML metamodel. This tool allows the modeling of the static diagrams and the dynamic sequence diagram defined in the language. It was developed as an Eclipse plugin through model driven approach and provides the mechanisms for model checking. However, the tool does not allow the complete modeling of the statics elements that compose a norm: only the modeling of permission and obligation norms for agents that are associated with agent roles, sub-organizations and organizations are supported.

Considering that MAS-ML tool and NorMAS-ML are based in MAS-ML metamodel, this makes it more suitable to be evolved in order to allow modeling of the norm diagram along with the model checking.

3 BACKGROUD

3.1 NorMAS-ML

NorMAS-ML (Freire et al., 2012) is a modeling language based in UML that allow the modeling of the all static elements that compose a norm (Figueiredo, 2011) along with the typical entities of MAS. This language is result of the MAS-ML extension (Silva et al., 2007).

The NorMAS-ML metamodel (Figure 1) was defined through the creation of the new metaclasses and relationships in MAS-ML metamodel in order to represent the following static normative elements defined by Figueiredo (2011):

- Deontic Concepts: the deontic logic refers to the logic of requests, commands, rules, laws, moral principles and judgments (Meyer and Wieringa, 1993). In MAS, such concepts have been used to describe the constraints for the behavior of agents in the form of obligations (what the agent must execute), permissions (what the agent can execute) and prohibitions (what the agent cannot execute).
- Involved Entities: considering that the norms are defined to restrict the entities behavior, the identification of the related entities is essential. The norm may regulate the behavior of individuals (for example, a particular agent, or an agent while playing a particular role), or the behavior of a group of individuals (for example, all agents playing a particular role, groups of agents, groups of agents playing roles or all agents in the system).
- Actions: once a norm is set to restrict the behavior of the entities, it is important the clear specification of the actions that are being regulated. Such actions may be communication, usually represented by sending and receiving a message, or non-communicative actions (such as access and modify a resource, get in an organization, move to another environment, etc.).
- Activation Constraints: a norm have a period of time in which its restrictions must be fulfilled, but only when this norm, is active. Norms may be activated by a constraint or a set of constraints that can be: the execution of actions, the definition of specific time intervals (before, after or in between), the reaching of system states or temporal aspects (such as dates) and also the activation/deactivation of other norm and fulfillment/violation of a norm.
- Sanctions: when a norm is violated the entity may suffer a punishment, and when a norm is fulfilled, the entity involved may receive a reward. Rewards and punishments are referred to as sanctions and should be related to the norm specification.
- Context: the norms are usually defined in a determined context that determines its application area. The norm may, for example, be described in the context of a specific environment and must be filled only by agents in execution in the environment. Similarly, a norm can be defined in the context of an organization and fulfilled only by the agents that play a role in the organization.

3.2 MAS-ML Tool

MAS-ML tool (Feijó, 2012) is a modeling environment that allows the modeling of MAS. Through this environment, developers can work with the concepts of problem domain, while using explicitly concepts defined in the solution domain, in this case the concepts and abstractions for the

paradigm of MAS.

Currently, the tool allows the modeling of all static diagrams: (i) Class (Farias et al., 2009), (ii) organization (Gonçalves et al., 2011) and (iii) role (Feijó, 2012). In addition, only the dynamic diagram of sequence is supported by the tool (Feijó, 2012).

The environment was developed as a plugin of the Eclipse platform (Eclipse Platform, www.eclipse.org, 2012). This implies that users can work with modeling of MAS while making use of the resources offered by the Eclipse platform. Since many agent platforms are implemented in Java, such as Jade (Bellifemine et al., 2007), Jadex (Pokahr et al., 2003), Jason (Bordini et al., 2007), the use of the Eclipse platform also facilitates a possible code generation within the same development environment.

4 EVOLVING MAS-ML TOOL

The version of the tool (Feijó, 2012) used as the basis for its evolution provides the modeling of all static diagrams defined in MAS-ML: diagram of classes, organization and roles. In other hand, the norm diagram from NorMAS-ML need to be defined in MAS-ML tool. Additionally, with the inclusion of norms in the tool, it is necessary to incorporate new rules that allow checking of conflicts in the definition of norms.

The strategy adopted to evolve the tool follows a model-driven approach, in which the metamodel of NorMAS-ML is used as a start point (Figure 1). In following are described in summary form, the six steps undertaken in the process of evolution of the tool:

- **Domain Model Extension:** were included metaclasses and stereotypes defined in NorMAS-ML, as well as the necessary adjustments in existing metaclasses.
- **Graphic Model Extension:** The new metaclasses of NorMAS-ML (Figure 1) were added in the Graph Model Definition and its relationships with other entities were created. Moreover, adaptations in existing metaclasses been incorporated.
- **Tool Model Extension:** new elements were created in the palette of the tool to create norms and their relationships required to build the norm diagram.
  - **Graphic Model Definition Extension:** at this stage, were created compartments for the representation of norms and their relationships (Figure 2), beyond the representation of nodes and

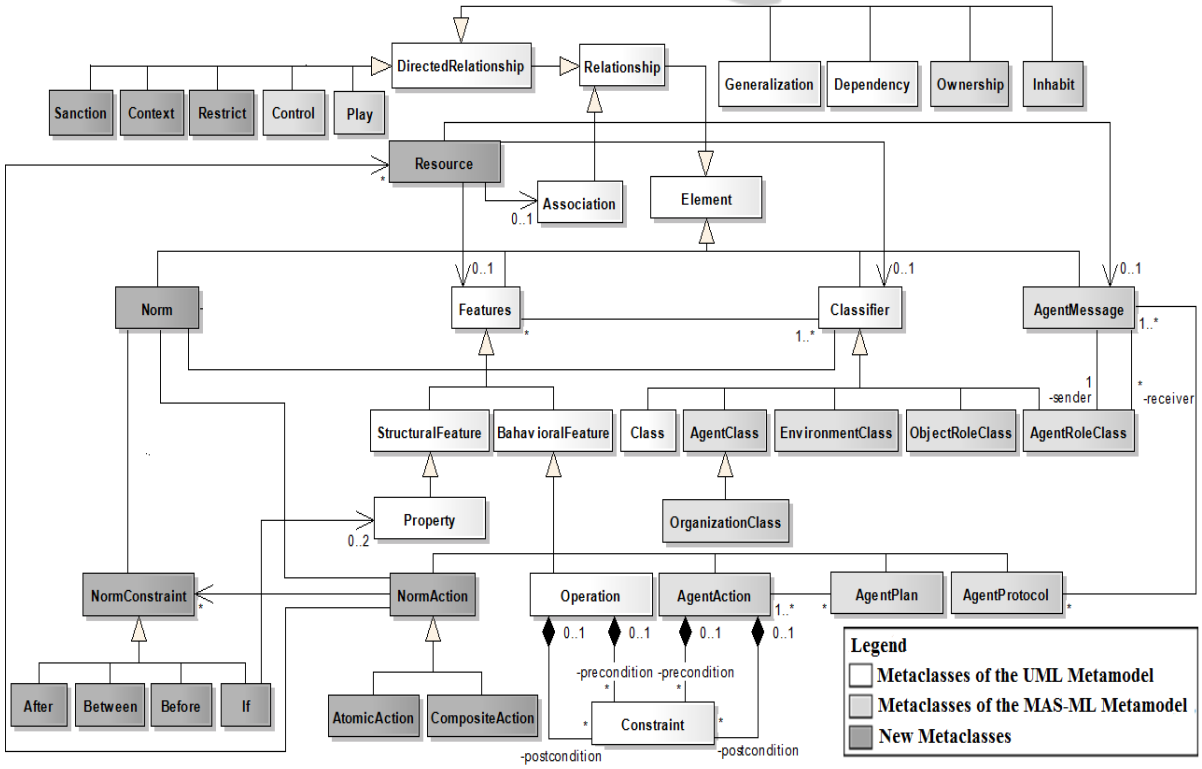

Figure 1: NorMAS-ML Metamodel (Freire et al., 2012).

connections from the previous step. Additionally, the graphic element of the entity AgentRoleClass was changed due to the removal of stereotypes <<duty>> and <<right>>. Similarly, the graphic element of OrganizationClass was changed due to the removal of the stereotype <<axiom>>.

- **Mapping Model Extension:** This model is created from a combination of the models defined in the previous steps, including validation rules to verificate the correctness of the models generated by constraints in OCL (OCL, 2012).

Finally, following the generative approach (Czarnecki and Eisenecker, 2000), the code of the tool is generated based on the extended platform specific model. Figure 3 shows the overview of the diagram editor for norms.

5 CASE STUDY

To illustrate the use of the new version of MAS-ML tool presented in Section 4, we used the Conference. Management System. This case study has been used by several authors, as Figueiredo (2011), Zambonelli et al. (2001), Dignum (2004) and Harmon et al. (2008), to illustrate their approaches.

The conference management systems are used to select the papers to be published in a scientific event. For this, authors should submit your papers before a certain date (deadline), from which the evaluators (at least three) initiate the review process. After the end of review period, event organizers must disclose the results.

Figure 2: Graphic elements for entities: (i) Norm, (ii) AgentRoleClass and (iii) OrganizationClass, and the relationships: (iv) Context, (v) Restrict, (vi) Reward and (vii) Punishment.

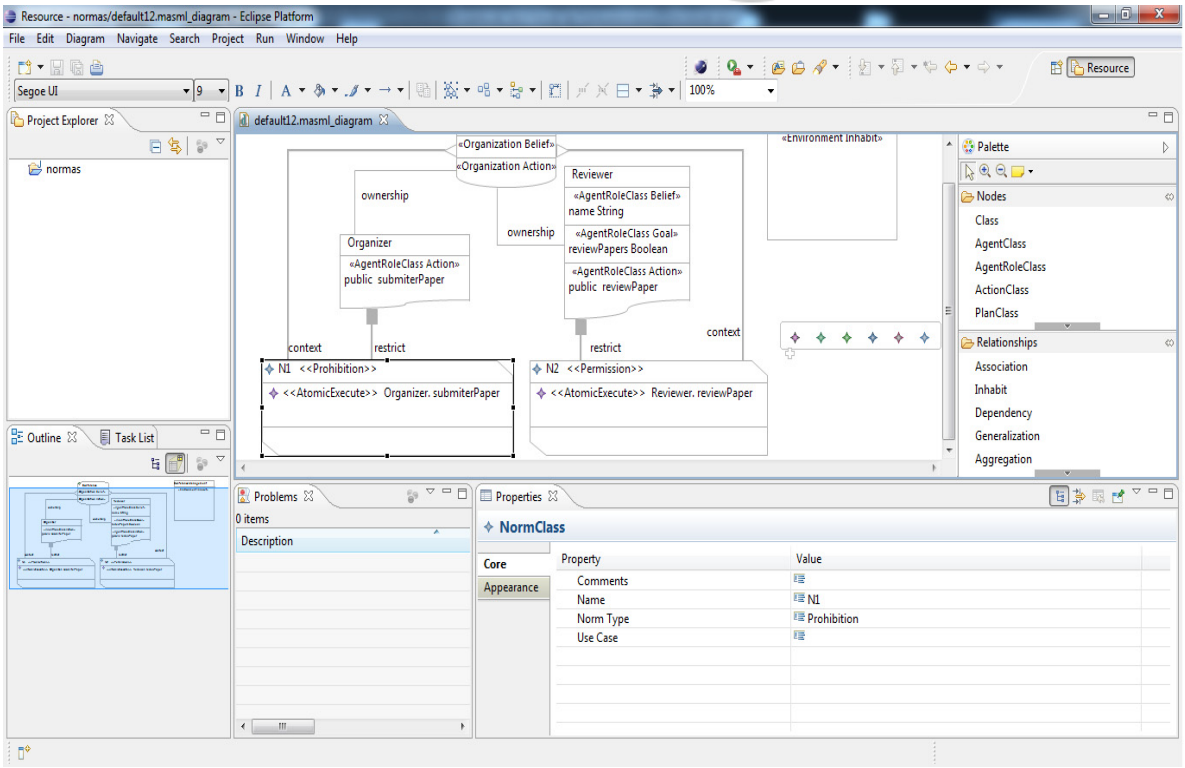

Figure 3: Overview of the norm diagram implemented in MAS-ML tool.

Authors can: (i) submit your papers (full or short) until the submission date, (ii) check the status of your paper and (iii) view review of the evaluators. Evaluators may: (i) submit papers and (ii) evaluate the papers that were listed by event organizers. The organizers may (i) extend the submission period of papers, (ii) choose the evaluators and (iii) disclose the results of the review.

5.1 Identification of the System's Entities

In the *Conference Management* environment we can identify the main organization *Conference* and the agent type: *user agent*, that can play the roles: *author speaker, organizer, conference chair, website manager* and *reviewer*. These roles are defined by main organization along with object role *submitted*. The instances of this role are played by instances of the class *Paper*, which has two subclasses: *ShortPaper* and *FullPaper*. Figure 4 shows the class diagram, specifying the relationships between classes and environments. Figure 5 illustrates the role diagram identifying the roles played by agents and objects in the organization *Conference*. Figure 6 shows the organization diagram referring to *Conference*. Thus, we define the classes of the organization along with main classes of agent, object, and agent role and object role.

5.2 Norms for System

For the conference management system, the following norms are defined:

- N1: The organizers are prohibited from submit papers.
- N2: Reviewers are allowed to submit papers.
- N3: The reviewers are prohibited from reviewing their own papers.
- N4 (Punishment for violation of N3): The reviewers that violate N3 should have their role canceled.
- N5 (Punishment for violation of N3): The president of the conference is obliged to drop paper.

Figure 7 shows the modeling of the norms N1 and N2. In addition, Figure 8 shows the modeling of the norms N3, N4 and N5.

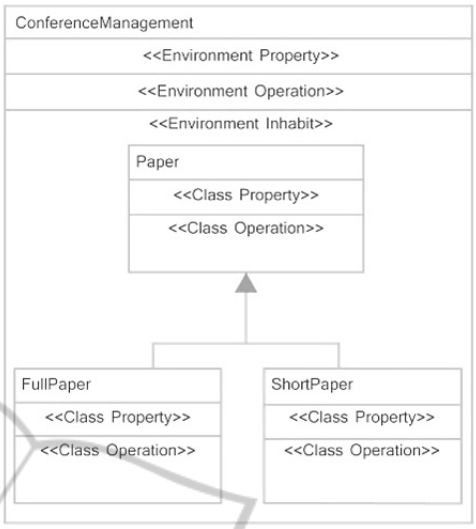

Figure 4: Class Diagram.

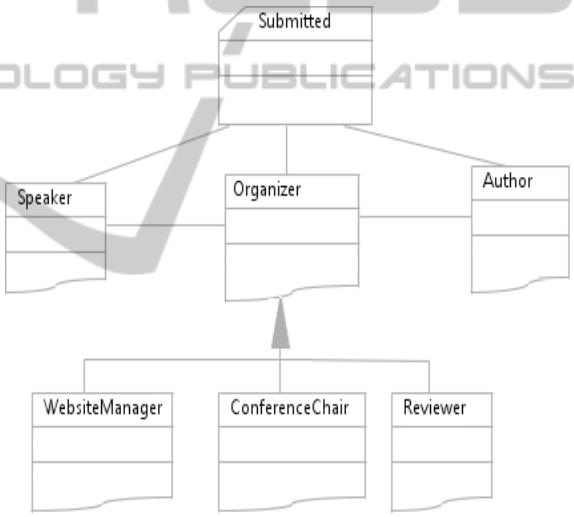

Figure 5: Role Diagram related to organization Conference.

6 CONCLUSIONS AND FUTURE WORKS

This paper presents the evolution of MAS-ML tool in order to support the modeling of normative multi-agent systems through the norm diagram defined by Freire et al. (2012) along with the model checking based on the NorMAS-ML metamodel. Thus, structural errors can be detected in advance, at design time, ensuring good formation of the models generated within the tool.

The approach to the evolution of MAS-ML tool is model-based where, graphic elements of the entity *Norm* and its relationships *Context, Restrict* and

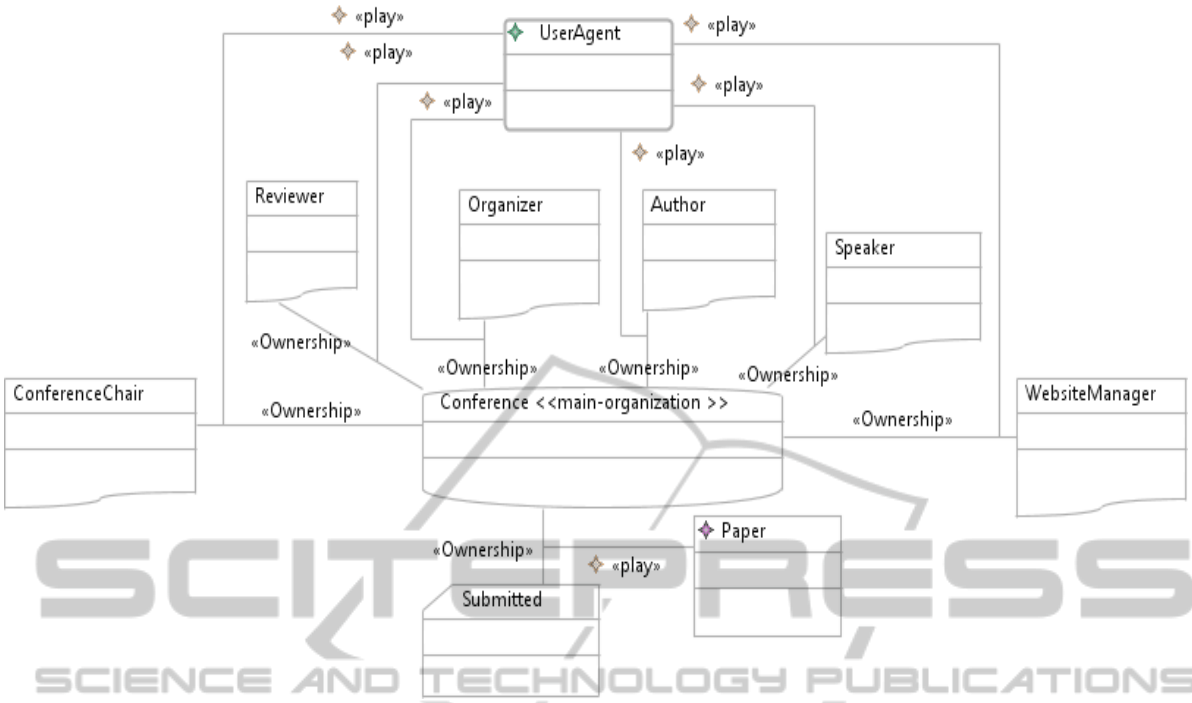

Figure 6: Organization Diagram represents the main organization Conference.

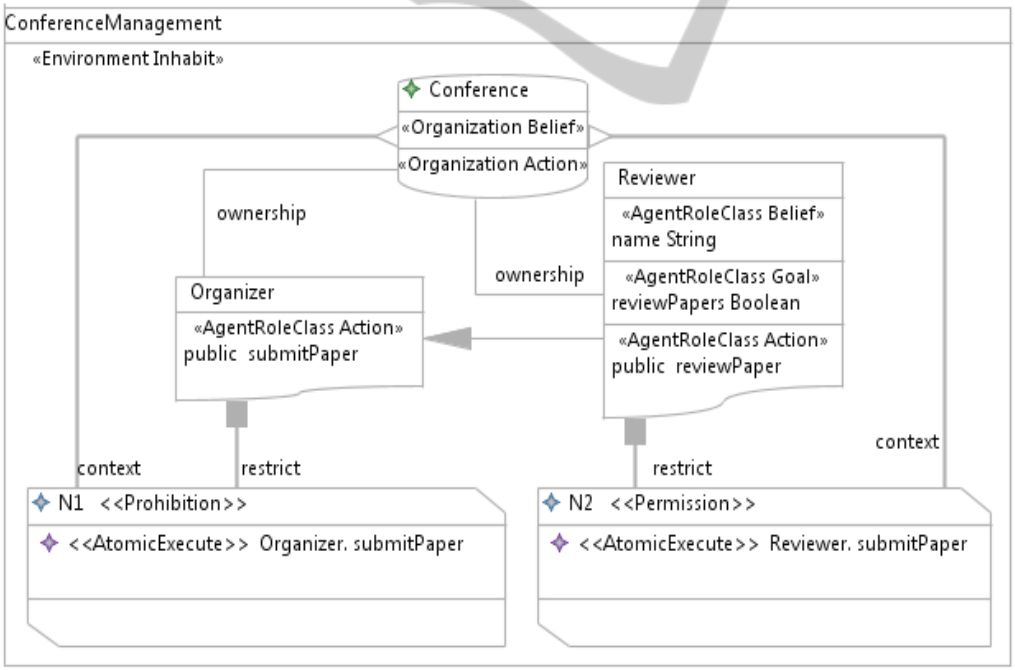

Figure 7: Modeling of norms N1 and N2 presented through the norm diagram.

Sanction (*Reward* and *Punishment*), defined in NorMAS-ML, were incorporated into the tool, and the entities *AgentRoleClass* and *OrganizationClass* had their graphic elements changed as the new definition in NorMAS-ML. Thus, in its new version, the tool supports the modeling of the entities that compose the normative multi-agent systems and their properties.

As future work, the checking of conflicts between norms can be cited. For this, the approach

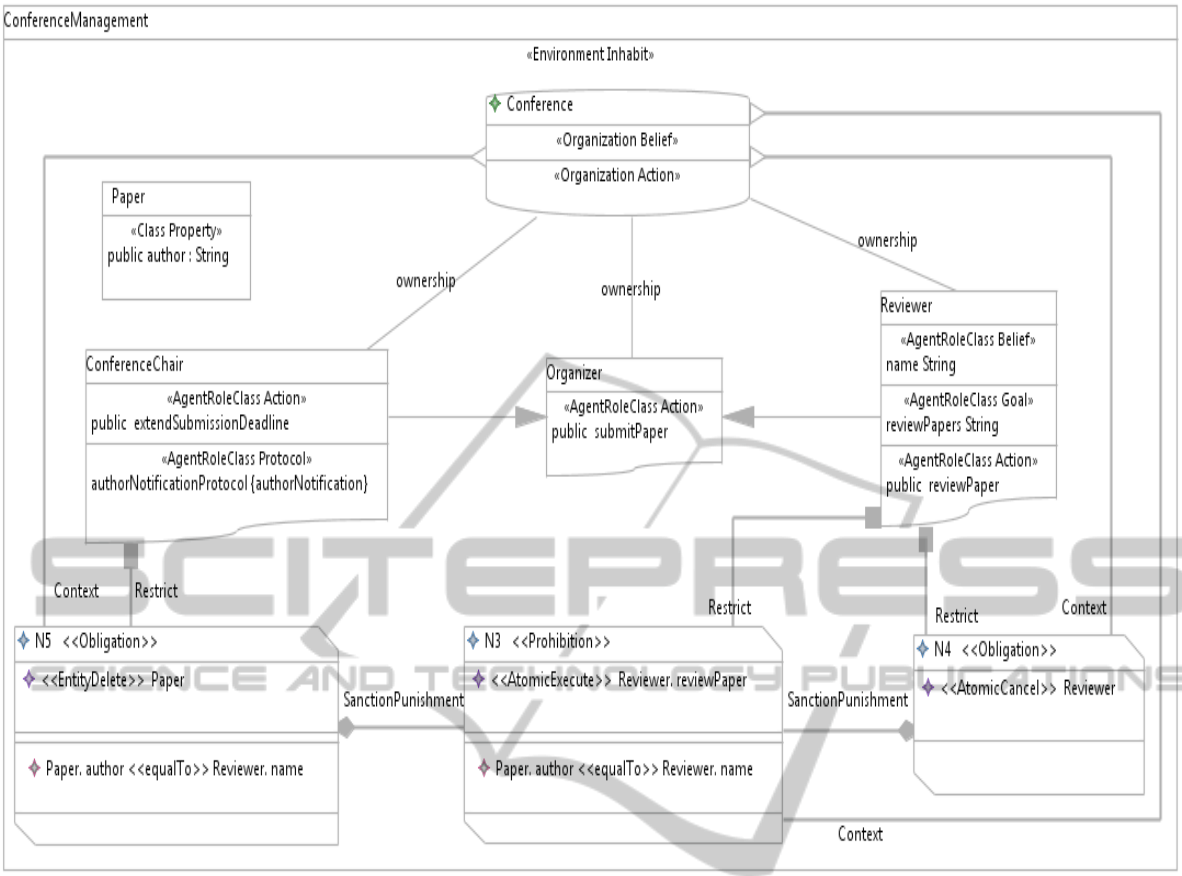

Figure 8: Modeling of norms N3, N4 and N5 presented through the norm diagram.

proposed by Figueiredo (2011) may be used. In addition, the code generation from diagrams generated by the tool can also be cited. For this task, a model-based approach similar to that proposed by Lopes et al. (2012) may be used.

REFERENCES

Bellifemine, F. L.; Caire, G.; Greenwood, D. (2007). Developing Multi-Agent Systems with JADE. [S.l.]: Wiley (Wiley Series in Agent Technology).

Bordini, R. H.; Wooldridge, M.; Hübner, J. F. (2007). Programming Multi-Agent Systems in AgentSpeak using Jason (Wiley Series in Agent Technology), John Wiley & Sons.

Czarnecki, K.; Eisenecker, U. (2000). Generative Programming - Methods, Tools, and Applications, Adison-Wesley, June 2000.

De Maria, B. A.; Silva, V. T.; Lucena, C.J.P.; Choren, R. (2005). VisualAgent: A Software Development Environment for Multi-Agent Systems. *Proceedings of the 19º Simpósio Brasileiro de Engenharia de Software* (SBES 2005), Brazil.

Dignum, V. (2004). A model for organizational interaction: based on agents, founded in logic. PhD dissertation, Universiteit Utrecht, SIKS dissertation series 2004-1.

Eclipse Platform (2012), http://www.eclipse.org/, acessado em 20 de Março de 2012.

Farias, K.; Oliveira, K.; Nunes, I.; Silva, V. T.; Lucena, C.J.P. (2009). MAS-ML Tool: Um Ambiente de Modelagem de Sistemas Multi-Agentes, V *Workshop on Software Engineering for Agent-oriented Systems* (SEAS 2009), Fortaleza, Brasil, pp. 1-12.

Feijó, A. R. (2012). Evolução da Ferramenta MAS- ML tool para a Modelagem dos Diagramas de Papéis e Sequência. Monografia. Fortaleza: *UECE, Departamento de Computação.*

Figueiredo, K. (2011). Modeling and Validation Norms in Multi-Agents Systems. Dissertação de Mestrado. Niterói: UFF, Instituto de Computação.

Figueiredo, K.; Silva, V. T. (2010). NormML: A Modeling Language to Model Norms. In: *1st Workshop on Autonomous Software Systems*. Salvador, Brazil.

Freire, E. S. S.; Cortés, M. I.; Gonçalves, E. J. T.; Lopes, Y. S. (2012). A Modeling Language for Normative Multi-Agent Systems. In: 13th *International Workshop on Agent-Oriented Software Engineering* (AOSE@AAMAS), Valencia (Spain).

Gonçalves, E. J. T.; Farias, K.; Cortés, M. I.; Feijó, A. R.; Oliveira, F. R.; Silva, V. T. (2011). MAS-ML TOOL - A Modeling Environment for Multi-agent Systems, In: *The International Conference on Enterprise Information Systems* (ICEIS 2011), Beijing, China.

Harmon, S. J.; Deloach, S. A. (2008). Trace-based Specification of Law in Guidance Policies for Multiagent Systems. Engineering Societies in the Agents World VIII, Springer-Verlag Berlin, Heidelberg.

Jennings, N. R. (1996). Coordination Techniques for DAI. In: Foundations of distributed artificial intelligence. Editores Greg O"hare; Nicholas Jennings. John Wiley and Sons.

Lopes, Y. S. ; Gonçalves, E. J. T. ; Cortés, M. I. ; Freire, E. S. S. (2012). A MDA Approach Using MAS-ML 2.0 and JAMDER. In: 13th *International Workshop on Agent-Oriented Software Engineering* (AOSE@AAMAS), 2012, Valencia (Spain).

OCL (2012), disponível em:<http://www.eclipse.org/modeling/mdt/?project=ocl>, acessado em 20 de Março de 2012.

Pokahr, A.; Braubach, L.; Lamersdorf, W. (2003). Jadex: Implementing a BDI-Infrastructure for JADE Agents. EXP - In Search of Innovation (Special Issue on JADE), vol. 3, no. 3 , Telecom Italia Lab, Turin, Italy, S. 76-85.

Silva, V. T.; Choren, R.; Lucena, C. J. P. de (2007). MAS-ML: A Multi-Agent System Modeling Language. *Conference on Object Oriented Programming Systems Languages and Applications* (OOPSLA); In: Companion of the 18th annual ACM SIGPLAN Conference on Object-oriented programming, systems, languages, and applications; Anaheim, CA, USA, ACM Press, pp. 304- 305.

Silva, V.; Cortés, M.; Lucena, C. (2004). An Object-Oriented Framework for Implementing Agent Societies. MCC32/04. Technical Report, PUC-Rio. Rio de Janeiro, Brazil.

Sommerville, I. (2007). Engenharia de Software. 8 ed. São Paulo: Pearson Addison-Wesley.

Meyer, J. J.; Wieringa, R. J. (1993). Deontic logic in computer science: normative system specification. Deontic logic in computer science: normative system specification, John Wiley and Sons Ltd. Chichester, UK.

Zambonelli, F.; Jennings, N. R.; Wooldridge, M. J. (2001). Organizational Rules as an Abstraction for the Analysis and Design of Multi -Agent Systems. In: *International Journal of Software Engineering and Knowledge Engineering, Volume 11*, Number 3.