An Approach to Regulate Open Multi-Agent Systems Based On A Generic Normative Ontology

Carolina Howard Felicíssimo and Carlos José Pereira de Lucena

Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), Rio de Janeiro, Brasil
Caixa Postal 38097 – 22.453-900 – Rio de Janeiro – RJ – Brazil

{cfelicissimo, lucena}@inf.puc-rio.br

Abstract. *In open multi-agent systems, unknown agents (implemented by different developers) can get in and get off environments according to their plans. Those agents are self-governed autonomous entities, pursuing their own individual goals based only on their own beliefs and capabilities. To avoid the risk of undesirable situations, mechanisms to regulate agents in open systems are needed. This paper presents a generic normative ontology as a regulatory mechanism for open multi-agent systems. This ontology provides a semantic support for agents to reason about environments' laws, organizations' laws and roles' laws.*

1. Introduction

Agents are autonomous flexible entities playing roles in order to meet their design objectives [Jennings 2000] – are, in short, goal-oriented entities. Agents inhabit environments, computational infrastructures to provide the conditions under which the entities that live in it exist. An agent can inhabit just one environment at the same time, but agents with the mobility characteristic can move from one environment to another, obeying or not the environments' laws. Agents are also adaptive entities, i.e. they can react to changing environments.

Environments can be represented either as centralized or distributed systems. In a centralized environment, all agents have access to the same structure. In a distributed environment, each cell behaves like a centralized environment in miniature [Ferber 1999]. Multi-agent systems (MAS) are typically open and have no single centralized designer [Huhns and Stephens 1999]. Key characteristics of open MAS are agent heterogeneity, conflicting individual goals and limited trust [Artikis et al. 2002]. The risk that agents will not be compliant with laws imposes the need of some regulatory mechanisms.

This paper aims at defining a mechanism to regulate open MAS based on an ontology. This ontology is a generic asset that needs to be instantiated and even extended for specific domains. An approach to design and built specific domain ontologies defines regulations with semantic support. This paper is organized as follow. Open multi-agent systems are briefly discussed in Section 2. The generic normative ontology is described in Section 3. A case study for the urban traffic domain is explained in Section 4, where the generic ontology was instantiated and extended. Finally, the conclusion is presented in Section 5.

2. Open Multi-Agent Systems

In multi-agent systems, agents interact usually with some intended individual or collective purpose. In **open** MAS, agents are self-governed autonomous entities that pursue their own individual goals based only on their own beliefs and capabilities [Abdelkader 2003]. Those agents are unknown entities (implemented by different developers), fact that increase the number of unpredictable and undesirable situations. To avoid that, it is necessary to structure a set of norms or conventions that articulate or restrain interactions in order to make them more effective in attaining those goals, more certain for participants or more predictable.

The use of regulatory mechanisms to control the behavior of agents can avoid the chaos that open MAS can become and also guarantee that agents will interact to collaboratively provide services. Regulatory mechanisms guiding the behaviors of agents, especially when their behaviors affect other agents and even their environments, provide some trust to the open system. However, in open MAS, enforcement has to be done over agent roles instead of agent's instances where their internal aspects are inaccessible. Also, it must be performed by the environment or by the environment's organizations where agents inhabit. Regulatory mechanisms can not simply be implicitly represented by constraints in agents' architectures or external fixed rules.

Norms are characterized by their prescriptiveness, sociality and social pressure [López et al. 2005]. They can be a regulatory mechanism while policing actions played by agents committed with roles. Roles are being represented as a fundamental concept in most knowledge representation languages, mainly in the design of open software systems.

3. A Generic Normative Ontology

Norms, while regulating environments, organizations and agents' roles, can control the actions performed in an open MAS defining which are permitted, obligated and prohibited. A permitted norm defines that an act is allowed to be performed; an obligatory norm defines that an act must be performed; a prohibited norm defines that an act must not be performed. The three types of norms described represent the three fundamental deontic statuses of an act [Alberti et al. 2005] from Deontic Logic [Wright 1951]. Deontic Logic enables to address the issue of explicitly and formally defining norms and dealing with their possible violation.

The three basic deontic notions of permission, obligation and prohibition are logically connected as presented by the following statements [Wright 1951]:

- If an act is permitted, then it is not prohibited.
- If an act is obligatory, then it is permitted and it is not prohibited.
- If an act is prohibited, then it is not obligatory and it is not permitted.
- If is not permitted not perform an act, then this act is obligatory.
- If is prohibited not perform an act, then this act is obligatory.
- If is permitted or is obligatory not perform an act, then this act is prohibited.
- If is permitted perform and not perform an act, then this act is permitted.

Agents as autonomous normative entities are able to take into account the existence of social norms in their decisions (either to follow or violate a norm) and are able to react to norms violations by other agents [Castelfranchi et al. 1999]. To provide norms regulation according to the Deontic Logic and consciousness of agents on MAS, a semantic support is desired. This type of support can be given by ontologies, making the represented information of a domain easier for machines to automatically process their meanings [McGuinness and Harmelen 2004].

Ontologies are conceptual models that embody shared conceptualizations of a given domain [Gruber 1993]. Ontologies languages are designed to be used by applications (machines) that need to process the content of information instead of just presenting information to humans [Smith et al. 2004].

Normative ontologies are those that have the norm concept as a central asset. This kind of ontology provides information for norm-autonomous agents, committed with their roles, to base their behaviors on, for example, goals and plans. A norm-autonomous agent capable of obtaining the semantic support provided by the normative ontology (its internal structure accesses ontologies) is an agent with intelligence for norm violation by an action selection mechanism. Action selection mechanisms [Tyrrell 1993; Blumberg 1997] permit effectiveness for agents to achieve their goals. Viruses are examples of agents that act non-compliant with norms, i.e. they are norm violation entities since they are compliant just with their own goals and plans.

A generic domain independent normative ontology is designed in order to illustrate regulations in open MAS. The ontology defines that the actions played in environments and organizations by agents committed with ther roles are restricted by norms. This ontology is constituted by six related main concepts (Figure 1[1]), all in the same hierarchical level: Environment, Organization, Role, Norm, Penalty and Action.

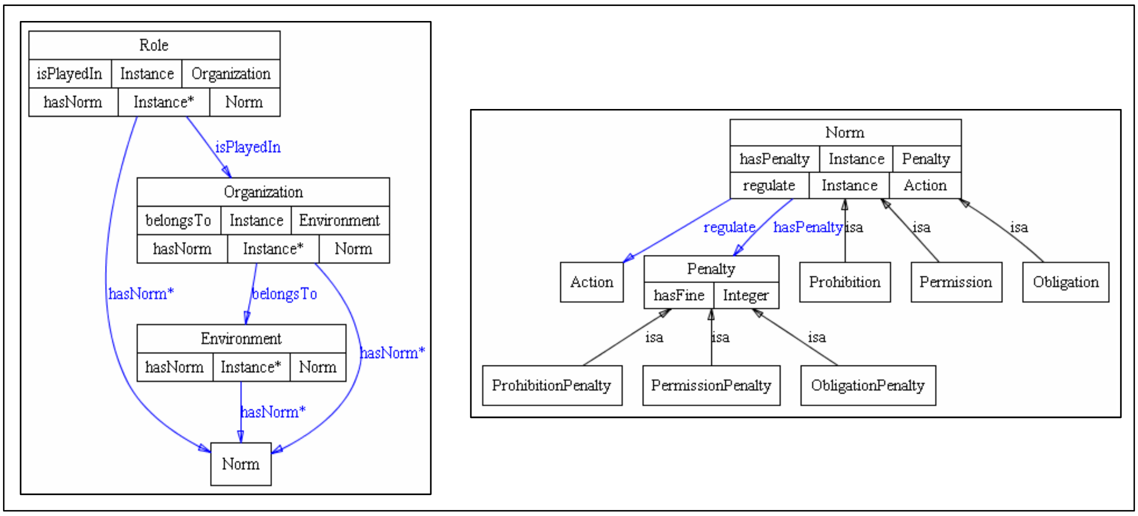

Figure 1. A generic normative ontology

[1] All ontologies presented in this paper were built by using the Protégé ontology editor [Protégé 2005] with its Ontoviz graph visualization plug-in [Ontoviz 2005]. In Ontoviz, concepts are related to their sub-concepts by the “isa” relationship and to their instances by the “io” (instance of) relationship.

This version of the generic normative ontology is an evolution from the one presented in [Felicíssimo 2005a], that is based just on roles. There, the concepts Environment and Organization were missed. Those concepts are essentials because they permit to define different roles' norms for organizations, different organizations' norms and different environments' norms, i.e. they permit to define more generic norms.

In the generic normative ontology designed, is possible to define norms for environments (the more generic defined norms), organizations and roles (the more specific defined norms) because they hold their associated norms. Because the use of specified levels of abstractions and the representation of the conceptual model by ontologies, it is possible to infer about inherit norms. In the role level, the norms have to be compliant with the norms inherited from the organization and the environment level. In the organization level, the norms have to be compliant with the norms inherited from the environment level.

The Environment concept holds its norms. The Organization concept holds its norms and the environment where the organization belongs to. The Role concept holds its norms and where those are valid (the organization where the role is played). The Norm concept holds its associated penalties, to inhibit norm violation, and holds its regulated actions. The Penalty concept holds a fine to be given if the associated norm is violated. The Norm and Penalty concepts are specialized in sub-concepts according to the Deontic Logic. The Action concept defines valid actions, meaning that actions not defined by the ontology will not make any effect in regulated organizations.

To be used in specific domains, the generic ontology needs to be instantiated and also can be extended. The following Section presents a case study of the urban traffic domain where the generic normative ontology was instantiated and extended.

4. Case Study

Urban Traffic Simulator System – UTSS – is an open multi-agent system with a semantic support to the governance of urban traffic laws in simulated environments. A simulated urban environment is a place where active entities move in it compliance or not with its laws. UTSS permits to analyze the behavior of agents while those play their roles in urban places.

An urban traffic's scenario, illustrated in Figure 2, is proposed as an UTSS instance. The proposed environment is composed (i) by agents playing the roles: car driver, pedestrian and police officer; (ii) by the four types of traffic signs: traffic signals, speed limit signs, one-way signs and two-way signs; (iii) by five numbered roadways and by sidewalks; and, finally, (iv) by the four places: church (accessed via path 1), house in the town (accessed via paths 2 and 4), house on the mountains (accessed via path 3) and hospital (accessed via path 5).

The paths of the environment have different characteristics. Paths 1 and 2 are in the same street, but with opposite directions. This street has a two-way sign and a speed limit sign. Those characteristics are the same for paths 3 and 4, but those paths also have a traffic signal close to the fork of the street. Path 5 is an emergency path to go to a hospital and has just a one-way sign.

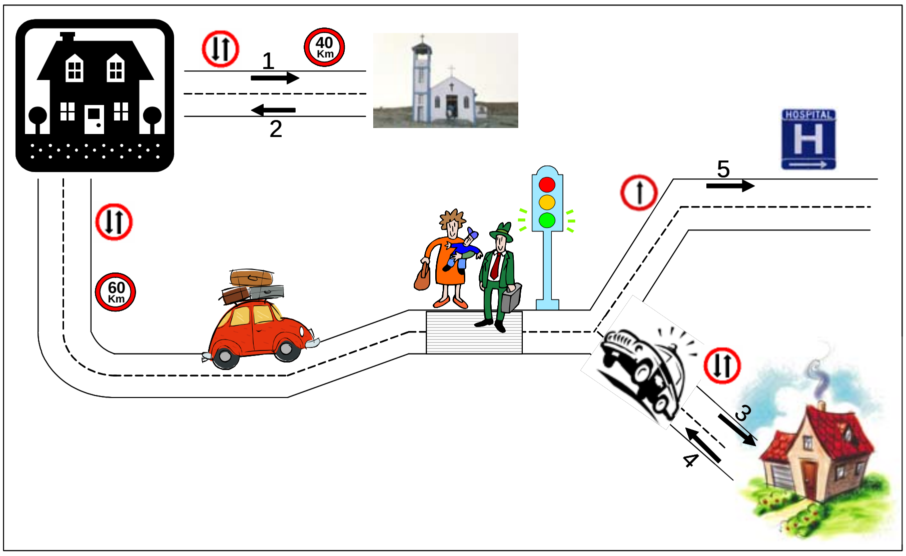

Figure 2. An urban traffic's scenario

Car drivers and pedestrians go and come from different places by using urban paths, which have several traffic signs. According to their goals, car drivers and pedestrians choose whether to obey or not the traffic signs rules. Police officers monitor the laws of traffic signs and can apply penalties for disobedient active entities. A traffic sign can have zero or more police officers monitoring it.

4.1. Ontology Extension

For the case study, in the role level the generic normative ontology illustrated in Figure 1 was extended by its Role and Action concepts, and by adding the Place concept. The extended ontology, illustrated in Figure 3, is called UTSS ontology.

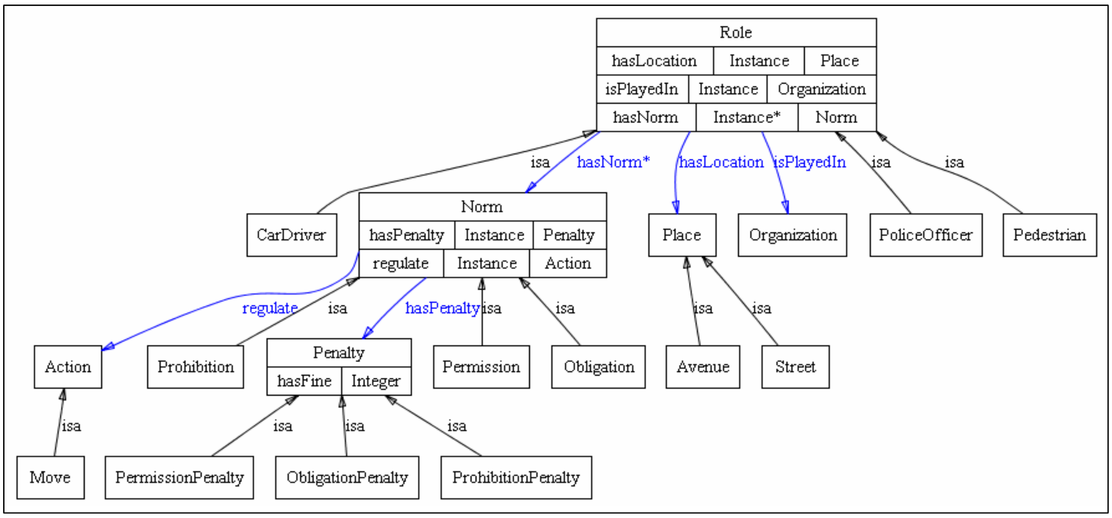

Figure 3. A normative ontology for the urban traffic domain, in the role level

The Role concept was extended by adding the *hasLocation* attribute to hold the place where the norms of the role are valid, and by adding the three sub-concepts: CarDriver, Pedestrian and PoliceOfficer. Those concepts represent examples of roles played in the urban traffic domain.

The Action concept was extended by adding the Move sub-concept with its four sub-concepts: MoveUnderSpeedLimit, MoveOnGreenLight, MoveInOneWay and MoveInTwoWays, representing respectively the traffic signs: speed limit, traffic signal, one-way sign and two-way sign. All the sub-concepts have each one a Boolean attribute with their same name, but changing the first letter by the same lower case letter. The MoveUnderSpeedLimit concept has also an integer attribute called *speed* to signalize its speed value.

The Place concept is added to the ontology with its two sub-concepts: Avenue and Street. Those types of places are specific urban paths' concepts.

4.2. Ontology Instantiation

To illustrate how agents can reason according to their norms with semantic support provided by ontologies, the UTSS ontology was instantiated by the concepts: Avenue, Street, CarDriver, Obligation, Permission, Prohibition, ObligationPenalty, PermissionPenalty, ProhibitionPenalty, MoveOnGreenLigth and MoveUnderSpeedLimit.

A role can have different norms depending on the place where it will be played. As an example, an avenue can have its speed limit greater than the one that a street has. Figure 4 illustrates the Place concept with its sub-concepts instantiated (in the figure, the *io* label means *instance of*). The Street concept was instantiated by the *Toneleros Street* (where are the paths 1 and 2 from the proposed environment) and the Avenue concept was instantiated by the *Sernambetiba Avenue* (where are the paths 3 and 4 from the proposed environment).

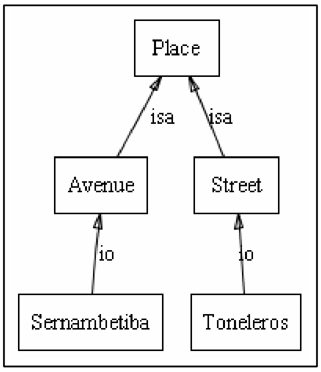

Figure 4. An example of avenue and street instances

For the car driver concept, two instances were created with the identifiers: *Mathew* and *William*. *Mathew* is driving in the *Toneleros Street* and *William* is driving in the *Sernambetiba Avenue*, both in the same organization called *Brazil*. The *Brazil Organization* belongs to an environment called *South America*, not illustrated. The car drivers are regulated by their norms. Figure 5 illustrates the car driver instances.

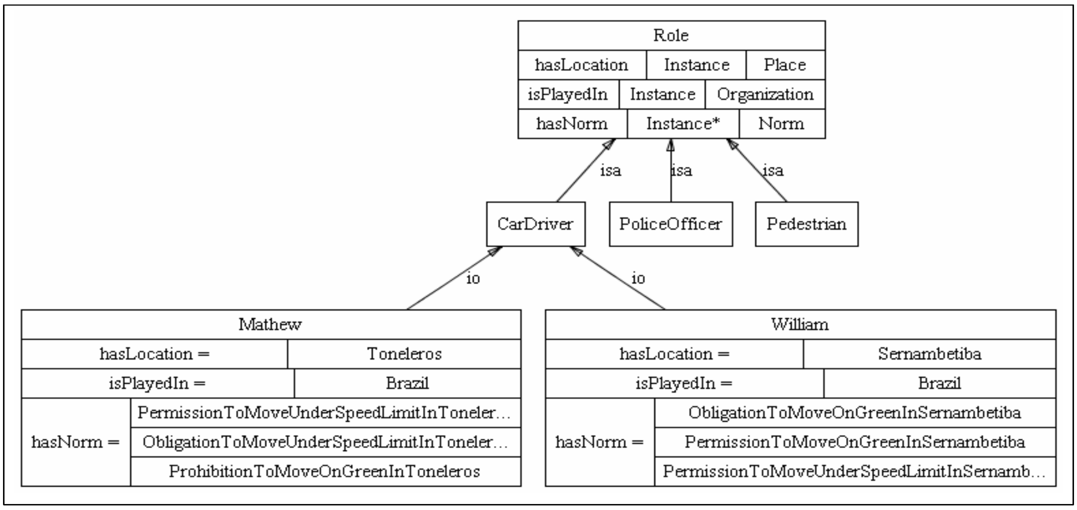

Figure 5. An example of car driver instances

In the UTSS ontology, roles have norms regulating actions. Figure 6 illustrates *William* (a car driver) playing in the *Sernambetiba Avenue* with his norms regulating the *MoveOnGreenInSernambetiba* action. *William* is regulated by his *PermissionToMoveOnGreenInSernambetiba* and *ObligationToMoveOnGreenInSernambetiba* norms that have the respectively penalties: *PermissionPenaltyToMoveOnGreenInSernambetiba* and *ObligationPenaltyToMoveOnGreenInSernambetiba*.

The *MoveOnGreenInSernambetiba* action, illustrated in Figure 6, has its *moveOnGreenLight* attribute set with the **true** value, meaning that *William* has the permission (*PermissionToMoveOnGreenInSernambetiba*) and obligation (*ObligationToMoveOnGreenInSernambetiba*) to move on the green light in the *Sernambetiba Avenue*. If *William* doesn't obey the law, other agents playing the police officer role can apply the *PermissionPenaltyToMoveOnGreenInSernambetiba* and the *ObligationPenaltyToMoveOnGreenInSernambetiba* penalties, totalizing a fine with its value equal to **15** (**5** from the first penalty and **10** from the second one). Agents playing the car driver role and with a high fine value can be banish from the environment.

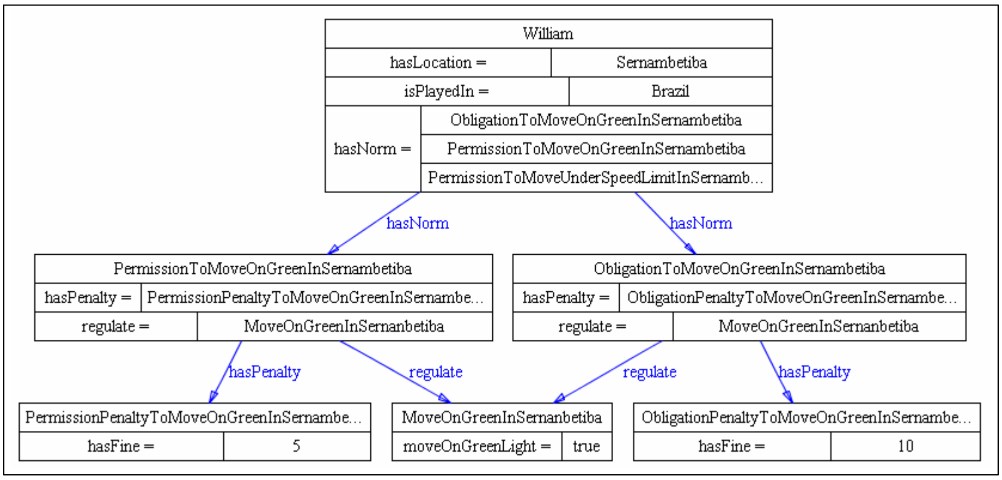

Figure 6. The *William* car driver with his norms regulating the *MoveOnGreenInSernambetiba* action

The *MoveUnderSpeedLimitInSernambetiba* action, illustrated in Figure 7, is regulated by the *PermissionToMoveUnderSpeedLimitInSernambetiba* norm, which has the *PermissionPenaltyToMoveUnderSpeedLimitInSernambetiba* penalty with its fine attribute's value equal to **1**. The action has its *moveUnderSpeedLimit* attribute set to **false** and its *speed* attribute set to **60**, meaning that *William* has the permission to move with his speed greater than **60** (the speed limit specified). This is an example where the speed limit sign addresses the minimum speed instead of the maximum speed.

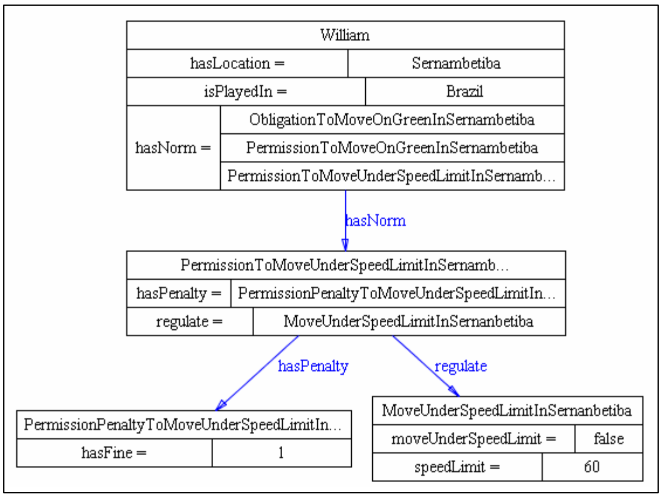

Figure 7. The *William* car driver with his norm regulating the *MoveUnderSpeedLimitInSernambetiba* action

Due the space limit, the UTSS ontology was not instantiated in the level of the concepts Environment and Organization, however, an example will be briefly discussed. For instance, the environment has the norm *ProhibitionToDisobeyAnyLaws*, meaning that agents are prohibited to disobey any laws from the environment. The organization *Brazil* has the norm *ObligationToMoveUnderSpeedLimit*, meaning that agents are obligated to move under the speed limit defined by the organization. An inconsistence will appear in the role level when the inherited norms will be combined with the norm *PermissionToMoveUnderSpeedLimitInSernambetiba*, illustrated in Figure 7. This norm regulates the *MoveUnderSpeedLimitInSernambetiba* action permitting agents to not move under the speed limit specified for the *Sernambetiba Avenue* (the action's *moveUnderSpeedLimit* attribute is set to **false**).

Specific norms have to be specified according to their more generic norms. Roles' norms have to be compliant with their organizations' laws. Organization's laws have to be compliant with their environments' laws. In that way, the inconsistence with the norms *ProhibitionToDisobeyAnyLaws*, *ObligationToMoveUnderSpeedLimit* and *PermissionToMoveUnderSpeedLimitInSernambetiba* would never have happened.

5. Conclusion

This paper presents a normative approach that uses ontologies to define regulations in open MAS. The generic normative ontology is independent of domain and it has six related main concepts: Environment, Organization, Role, Norm, Penalty and Action. This ontology structure provides semantic support for agents to base their behavior according to norms and to reason about action selection.

The generic normative ontology was instantiated and extended for a case study of the urban traffic domain. As a future work, the supply chain scenario will be explored by using the TAC SCM specifications [Collins 2005]. The results of others case studies will probably interfere in the current version of the generic ontology, which will probably need to be reviewed and also improved.

Acknowledgements

This work has benefited considerably from extensive discussions with Gustavo Carvalho, Rodrigo Paes, Viviane da Silva, Firmo Freire, Leonardo Cunha and Daniela Brauner, and from the comments in the embryonic works: [Felicíssimo 2005a] and [Felicíssimo 2005b], by the reviewers of the *2005 AAAI Fall Symposium on Roles, an interdisciplinary perspective*. The funding from CNPq as part of individual grants and the ESSMA project (552068/2002-0) is also acknowledged.

References

Abdelkader, G. (2003) Requirements for achieving software agents autonomy and defining their responsibility. In Proc. Autonomy Workshop at AAMAS.

Agha, G. A. (1997) Abstracting Interaction Patterns: A Programming Paradigm for Open Distributed Systems. In (Eds.) E. Najm and J.-B. Stefani, Formal Methods for Open Object-based Distributed Systems IFIP Transactions, Chapman & Hall.

Alberti, M.; Gavanelli, M.; Lamma, E.; Mello, P.; Torroni, P. and Sartor, G. (2005) Mapping Deontic Operators to Abductive Expectations. In Proc. of the 1st International Symposium on Normative Multiagent Systems (NorMAS 2005), UK.

Artikis, A.; Pitt, J. and Sergot, M. (2002) Animated specifications of computational socities. In (Eds.) C. Castelfranchi and W. Lewis Johnson, Proc. of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2002), Part III, pages 1053-1061, Italy.

Blumberg, B. M. (1997) Old Tricks, New Dogs: Ethology and Interactive Creatures. PhD Thesis from the program in media arts and sciences, school of architecture and planning, Massachusetts Institute of Technology.

Castelfranchi, C.; Dignum, F.; Jonker, C.M. and Treur, J. (1999) Deliberative Normative Agents: Principles and Architecture. In Proc. of the Sixth International Workshop on Agent Theories, Architectures, and Languages (ATAL-99), USA.

Collins, J; Arunachala,R; Sadeh,N; Eriksson,J; Finne,N; Janson,S. (2005) The Supply Chain Management Game for the 2005 Trading Agent Competition. CMU-ISRI-04-139. Available at: http://www.sics.se/tac/tac05scmspec_v157.pdf

Felicíssimo, C. H.; Lucena, C.; Carvalho, G. and Paes, R. (2005a) Normative Ontologies to Define Regulations Over Roles in Open Multi-Agent Systems. Extended abstract for: AAAI Fall Symposium on Roles – an interdisciplinary perspective, 2005, Hyatt Crystal City in Arlington, Virginia. To appear.

Felicíssimo, C. H.; Lucena, C. J. P. de and Silva, V. T. da. (2005b) From Roles Modeled by Using the MESSAGE Methodology to their Implementation with the ASF Framework. Extended abstract for: AAAI Fall Symposium on Roles – an interdisciplinary perspective, 2005, Hyatt Crystal City in Arlington, Virginia. To appear.

Ferber, J. (1999) Multi-Agent Systems, An Introduction to Distributed Artificial Intelligence. Addison-Wesley, ISBN: 0-201-36048-9, Great Britain.

Gruber, T. R. (1993) A translation approach to portable ontology specifications. Knowledge Acquisition, Vol. 5, Issue 2 (1993, June), pages 199-220, ISSN:1042-8143.

Huhns, M.N. and Stephens, L.M. (1999) Multi-Agent Systems and Societies of Agents. G. Weiss (ed.), Multi-Agent Systems, ISBN 0-262-23203-0, MIT Press.

Jennings, N. R. (2000) On Agent-Based Software Engineering. Artificial Intelligence, 117 (2) pages 277-296.

Lopéz, F. L. y; Luck, M. and d'Inverno, M (2005). A Normative Framework for Agent-Based Systems. In Proc. of the 1st International Symposium on Normative Multiagent Systems (NorMAS 2005), UK.

McGuinness, D. L. and Harmelen, F. van. (2004) OWL Web Ontology Language Overview. W3C Recommendation 10 February 2004. In <http://www.w3.org/TR/owl-features/>. Accessed in May, 2005.

Ontoviz (2005). Ontoviz - Allows to visualize Protégé ontologies with the help of a highly sophisticated graph visualization software. In <http://protege.stanford.edu/plugins/ontoviz/ontoviz_tab_bin.zip/>. Accessed in May, 2005.

Protégé (2005). Protégé - A free, open source ontology editor and knowledge-base framework. Current recommended release: version 3.0. In <http://protege.stanford.edu/download/release/full/>. Accessed in May, 2005.

Smith, M. K.; Welty, C. and McGuinness, D. L. (2005) OWL Web Ontology Language Guide. W3C Recommendation 10 February 2004. In <http://www.w3.org/TR/owl-guide/> . Accessed in May, 2005.

Tyrrell, T. (1993) Computational Mechanisms for Action Selection. PhD Thesis from University of Edinburgh.

Wright, G. H. von (1951). Deontic Logic. Mind, New Series, Vol. 60, No. 237 (Jan., 1951), pages. 1-15.