Supporting Regulatory Dynamics in Open MAS

Carolina Felicíssimo[1], Ricardo Choren[2], Jean-Pierre Briot[1, 3] and Carlos Lucena[1]

[1] DI, PUC-Rio, Rua Marquês de São Vicente, 225, Gávea
Rio de Janeiro, RJ, 22453-900, Brasil
{cfelicissimo, lucena}@inf.puc-rio.br
[2] SE/8 – IME, Pça Gen Tibúrcio 80, Urca
Rio de Janeiro, RJ, 22290-270, Brasil
choren@de9.ime.eb.br
[3] LIP6, 8 rue du Capitaine Scott
75015 Paris, France
jean-pierre.briot@lip6.fr

Abstract. An open multi-agent system should enable regulatory dynamics easily in order to provide updated information about the system enforcement for its law-aware agents. This work presents a case study where agents can dynamically adapt their behaviors according to current enforcements of contextual laws. The case study uses a flexible solution for contextual regulations, which includes a top-down contextual modeling of laws, a normative ontology to formalize the defined regulatory contexts and a rule support for law retrieval.

1 Introduction

Multi-agent systems (MAS) are a powerful technology that has emerged as a promising approach to develop information systems that are composed of several goal-oriented problem-solving entities [14]. Openness [10] has led to software systems that have no centralized control and that are formed of autonomous entities. In this work, we assume that an open MAS is a system that puts together a set of heterogeneous, self-interested agents whose actions may deviate from the expected behavior in a context. A regulated (or normative) open MAS provides norms that support oversight of the agents, so that the agents can be accounted for executing undesirable actions.

Normative agents are agents that have explicit knowledge about the enacted norms in MAS (the system laws) and can make a choice whether to obey the norms or not in specific cases. So, all agents must know (or be aware of) the actions, norms and penalties of the regulated system. A norm model should not only provide the means to formalize norm regulations, but also enable it to work with a flexible rule support mechanism to assist the current agent regulation during system execution. Besides, this mechanism should be easy to operate so that norms can be created, deleted and modified at run-time.

In this paper we describe a case study focusing in the regulatory dynamics of contextual laws. This case study is based on an approach for contextual regulations in open MAS, which supports deliberative normative agents with information about the

current system enforcement of laws. The approach [3, 4, 5], briefly presented in Section 2, uses a flexible solution for contextual regulations, including a top-down contextual modeling of laws, a normative ontology to formalize the defined regulatory contexts and a rule support for law retrieval. The case study is described in Section 3. Our approach for contextual regulations is compared to a related work in Section 4. Finally, we offer our conclusion and outline a future work in Section 5.

2 Law-Aware Open Multi-Agent Systems

Before we proceed to present our law-governance technique we begin by discussing some concepts. A MAS is constituted, mainly, by environments, organizations, agents, agent roles and agent interactions [13]. Environments [20] are discrete computational locations (similar to places in the physical world) that provide conditions for agents to inhabit it. Organizations [6] are social locations where group of agents play roles inside it, seeking to achieve their goals. Agent roles are abstractions that define a set of related tasks [18]. Agents interact with other agents, from the same or from different organizations and environments.

Environments, Organizations, Agent Roles and Agent Interactions suggest different contexts for regulations in MAS. Contexts can be defined as pieces of information which characterize the situation of participants [1]. Context-aware systems use contexts to provide relevant information and/or services to their users, where relevancy depends on the users' tasks [1]. In our definition, regulated context-aware systems use contextual law information to provide the current enforcement of laws to their users.

For regulations in open MAS, we are currently dealing with the four following contexts of laws: environment, organization, role and interaction. *Environment Laws* are applied to all agents from the regulated environment. *Organization Laws* are applied to all agents from the regulated organization. *Role Laws* are applied to all agents playing the regulated role, and finally, *Interaction Laws* are applied to all agents involved in the regulated interaction.

Regulatory contexts, i.e. contexts for enforcement of laws, should have their semantic expressed in a meaningful way. Ontologies [9], i.e. conceptual models that embody shared conceptualizations of a given domain, can represent information in a well-defined way for agents to process their contents. Thus, we developed a normative meta-ontology to represent our four regulatory contexts. This meta-ontology is illustrated in Fig. 1.

The environment, organization and role regulatory contexts of laws are represented in our meta-ontology by the six related main concepts, all at the same hierarchical level: Environment, Organization, Role, Norm, Penalty and Action. The interaction regulatory context of laws must be implemented during the ontology instantiation and extension processes by following the representation pattern from the Semantic Web Best Practices document [16]. This pattern defines that the relation object itself is represented by a created concept that will link the other concepts from the relation. In our approach, interaction laws are represented by a new sub-concept of the Norm concept linking Role concepts.

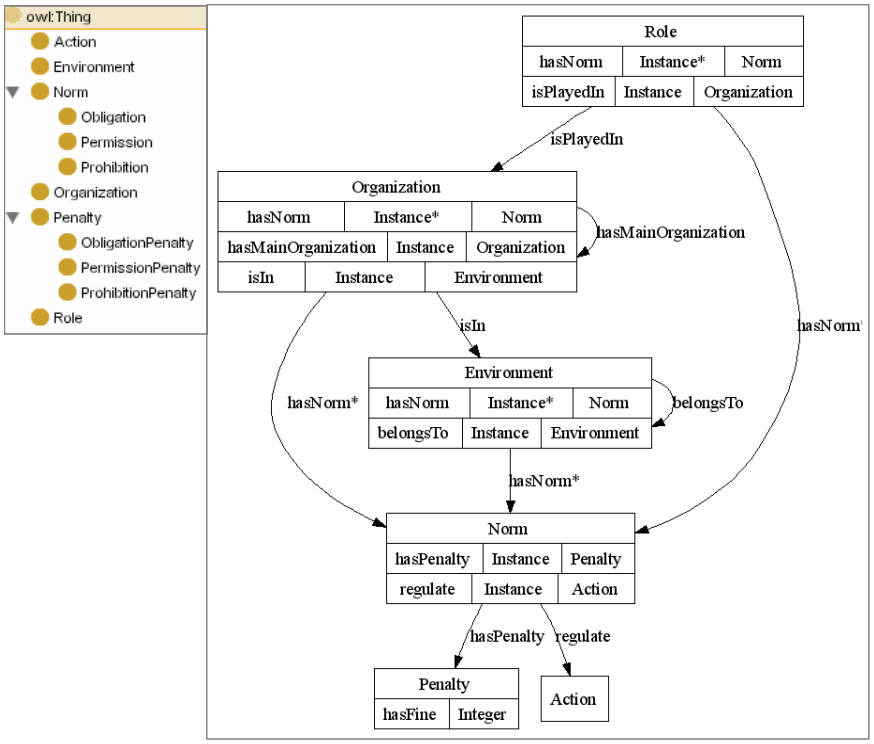

Fig. 1. Our normative meta-ontology

Laws of regulated open MAS can be represented by norms, which should in some way influence the behavior of its agents. Norm-aware agents use norm information to better adapt their behavior. Thus, norms should control environments, organizations, agent roles and agent interactions by defining which actions are *permitted*, *obliged* and *prohibited* to be executed. A *permitted* norm defines that an act is allowed to be performed; an *obliged* norm defines that an act must be performed; and a *prohibited* norm defines that an act must not be performed.

Nevertheless, norms cannot be incorporated into the agents, since we do not have control over their development. So, the regulatory mechanism should allow for some facility while applying norms. In this sense, the instantiated ontology should be complemented by a set of rules, which should be responsible for composing laws from different regulatory contexts and for informing the system and its agents of which laws they are bound to, at a given moment in time.

The main idea behind using a set of rules is to avoid implementations, in the regulated system and in its agents, of related codes for law compositions and retrievals, according to different regulatory contexts. Therefore, a rule inference engine reads the normative ontology instance (where laws are expressed in) and, based on the set of rules (where the contextual compositions of laws are defined in), it retrieves the laws by using inference. The process is illustrated in Fig. 2.

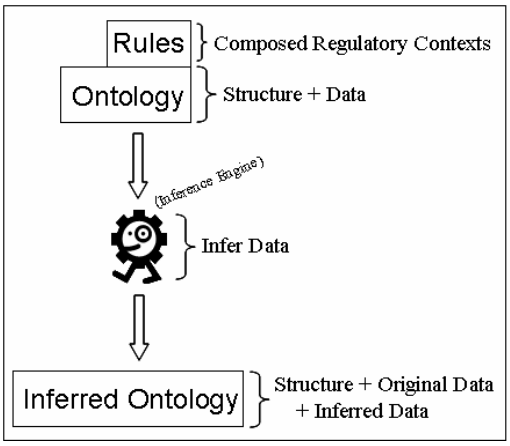

Fig. 2. Composing and retrieving contextual data (laws)

It is important to remark here that no implementation code is needed, either in the regulated open MAS or in its agents, to dynamically compose the regulatory contexts or to check out the ontology for law retrieval. This makes our solution more flexible since law management is done outside implementation codes.

Modular context refinements allow a more flexible system, providing a better support to manage regulatory dynamics. Our approach provides a simple way to manage law evolutions in two different cases. The first is when laws need to be added, updated or deleted to the regulated system. For this case, simply updating the modeling and ontology makes the evolution and, then, they are sanctioned in the system without the need to stop it. This happens because laws are written in the ontology, and the system and its agents always check it at run-time. The second case for law evolution is when the defined contextual compositions of laws need to be modified. For this case, the evolution is made by simply updating the set of rules, according to the new contextual regulations. Once this set is updated, the evolution is realized, because the system and its agents are checking the set of rules whenever they retrieve a law.

3 Case Study

The domain of multinational corporations will be used in order to explain why contextual regulations are an important issue for law enforcement in open MAS. A multinational corporation is an enterprise that manages production branches located in at least two countries. These branches can be in different regions across multiple continents. Corporate governance includes regulate all possible relationships among the many players involved.

We defined situations with two multinational corporations (organizations) specifically to present our case study. Our system's world is composed of seven environments (North America, South America, Canada, United States of America, Argentina,

Brazil and Chile), two main organizations (Dellie and Hpie); four organizations (Hpie Canada, Hpie Argentina, Dellie Brazil and Dellie Chile); and five roles (supplier, manufacturer, distributor, retailer and customer). All entities are illustrated in Fig. 3.

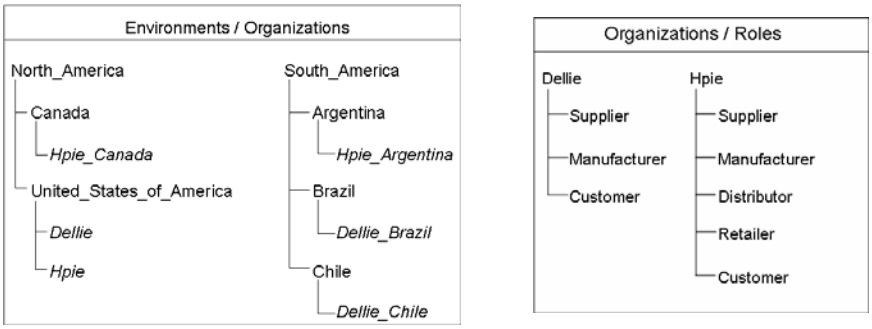

Fig. 3. Entities from our system's world

Normally, corporation laws are not public because they are strategically for the corporation businesses. Because of this, we created environment, organization, role and interaction laws based on some public laws collected from several corporate Web sites. The created laws were classified according to our four regulatory contexts.

1. Examples of Environment Laws:

1.1. In North America, a finished good from every organization has its price added with a percentage of the price value (dependent of the seller location) as taxes if the deliver is immediately (carry-on) or if the deliver address is in the seller location.

1.2. In Canada, a finished good from every organization has its price added with 15% of the price value as taxes if the deliver is immediately (carry-on) or if the deliver address is in Canada.

1.3. In the state of the Dellie headquarter (in United States of America), a finished good from every organization has its price added with 8% of the price value as taxes if the deliver is immediately (carry-on) or if the deliver address is in the state of the Dellie headquarter.

1.4. In the state of the Hpie headquarter (in United States of America), a finished good from every organization has its price added with 5% of the price value as taxes if the deliver is immediately (carry-on) or if the deliver address is in the state of the Hpie headquarter.

1.5. In South America, every finished good has in its price taxes included.

2. Examples of Organization Laws:

2.1. In the world, every Hpie organization has to follow the *direct sales to customer* model, i.e. sales of the organization's products can just be made between: suppliers and manufacturers, or manufacturers and distributors, or distributors and retailers, or retailers and customers.

2.2. In Hpie Argentina, sales of the organization's products can just be made between: suppliers and manufacturers, or manufacturers and distributors, or distributors and retailers, or distributors and customers, or retailers and customers.

2.3. In Dellie, just suppliers and manufacturers are permitted to sell the organization's products to customers.

2.4. In Dellie Chile, sales of the organization's products can just be made between: suppliers and manufacturers, or manufacturers and customers.

3. Examples of Role Laws:

3.1. In Dellie, customers can receive only complete orders.

3.2. In Hpie Canada, suppliers must ship orders on their due dates.

3.3. In Dellie Brazil, suppliers must ship orders until their due dates.

3.4. In Dellie Brazil, customers must receive orders until one day after their due dates.

3.5. In Hpie Argentina, customers must give a down payment of 10% to every placed order.

4. Examples of Interaction Laws:

4.1. In Dellie, manufacturers have the permission to pay in up to 30 days after they receive their orders from suppliers.

4.2. In Dellie Brazil, manufacturers have 10% of discount in the total price of their orders if the payment to their suppliers is done in cash.

4.3. In Hpie Canada, suppliers have the permission to ship incomplete orders to manufacturers.

We formalized all the presented laws by instantiating our normative meta-ontology. The ontology instance extended the meta-ontology with new concepts related to the representation of the interaction laws and the five roles (supplier, manufacturer, distributor, retailer and customer). Some parts of the ontology instance are illustrated in Fig.4, Fig.5, Fig.6 and Fig.7 with instances of related concepts.

The main idea of this case study is to outline both: the diverse possibilities to which agent actions can be regulated according to different contexts and how difficult can be the task of law enforcement. Our approach intends to provide support for law-aware agents to reason and to adapt their behavior according to the enforcement of the system laws. To exemplify this, two scenarios are given in the two following subsections: in the first scenario (subsection 3.1), customers will choose where to buy Hpie products based on the need to have their orders in due dates; in the second scenario (subsection 3.2), manufacturers will choose where to buy based on their wish to make good deals with suppliers.

Our approach also permits restriction and relaxation of laws, while composing laws from different regulatory contexts. To exemplify restriction of laws, a scenario is given in subsection 3.3; to exemplify relaxation of laws, a scenario is given in subsection 3.4. When laws from the same regulatory context are composed, some conflicts can be raised. To exemplify this, a scenario is given in subsection 3.5.

3.1. A Scenario where Customers Need their Orders in Due Dates

To exemplify how law-aware agents can adapt their behavior according to the current enforcements of contextual laws, a scenario is given by a customer in North America looking for Hpie products. In order to minimize deliver expenses, the customer can choose to buy in Hpie or in Hpie Canada (Hpie organizations in North America). These organizations with their relationships are illustrated in Fig. 4.

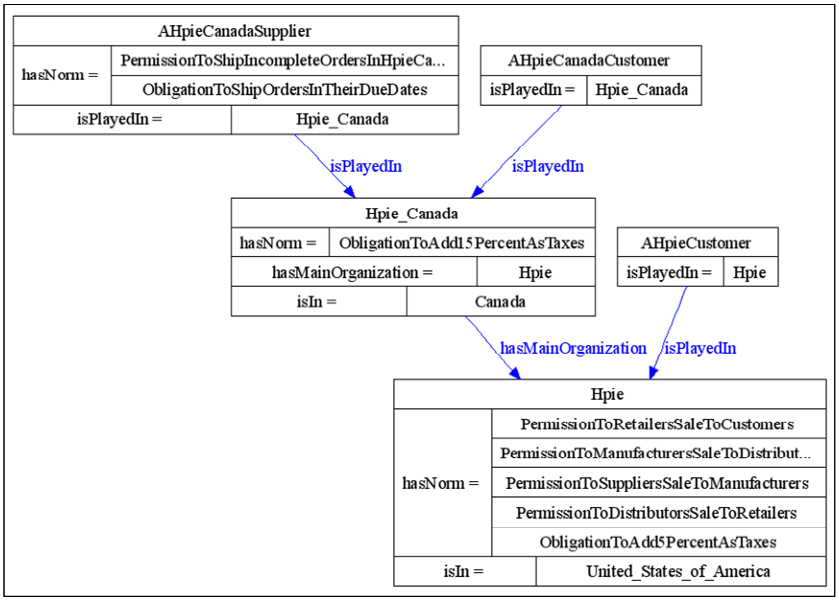

Fig. 4. Possible regulatory contexts for Hpie customers

If the customer decides to buy in Hpie (being "AHpieCustomer"), he is restricted to buy products just from retailers (organization law 2.1), but he pays only 5% of the price value as taxes if the deliver is immediately (carry-on) or if the deliver address is in the state of the Hpie headquarter (environment law 1.4).

If the customer decides to buy in Hpie Canada (being "AHpieCanadaCustomer"), he has to pay 15% of the price value as taxes if the deliver is immediately (carry-on) or if the deliver address is in Canada (environment law 1.2). In Hpie Canada, the customer can also buy direct from suppliers and, doing that, he has the guarantee that his orders will be shipped in their due dates (role law 3.2). However, if Hpie Canada is regulated with its main organization laws (Hpie organization laws) too, the customer is restricted to buy products just from retailers (organization law 2.1), but he can pay only 5% of the price value if the deliver is immediately (carry-on) or if the deliver address is in the state of the Hpie headquarter (environment law 1.4).

Because Hpie and Hpie Canada are organizations in North America, both can also be regulated by the North America environment law 1.1. This law is more general than the environment laws 1.2 and 1.4, and thus takes no effect to the regulation.

3.2. A Scenario where Manufacturers Look for Good Deals with Suppliers

For another example of how law-aware agents can adapt their behavior according to the current enforcements of contextual laws, a scenario is given by a manufacturer in North America looking for suppliers. In order to minimize deliver expenses, the manufacturer can choose to buy with Dellie, Hpie or Hpie Canada suppliers (North America suppliers). The Dellie organization, some of its roles and an interaction law are illustrated in Fig. 5.

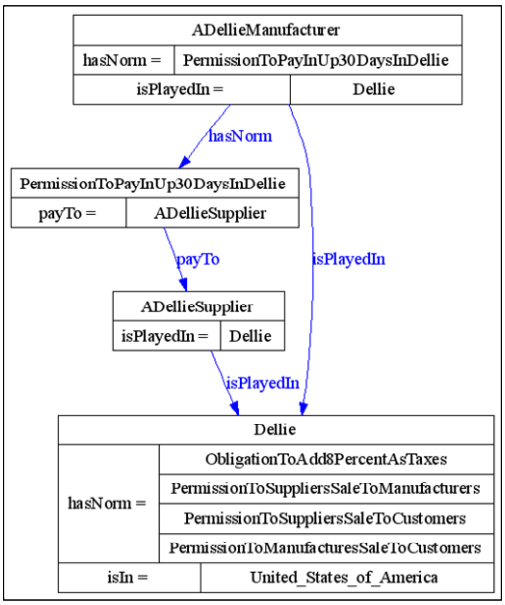

Fig. 5. Possible regulatory contexts for a Dellie manufacturer

If the manufacturer decides to buy in Dellie (being "ADellieManufacturer") with one of the Dellie suppliers, he has the benefit to pay in up to 30 days after he receives his orders (interaction law 4.1). Besides that, he pays 8% of the price value as taxes if the deliver is immediately (carry-on) or if the deliver address is in the state of the Dellie headquarter (environment law 1.3).

If the manufacturer decides to buy in Hpie Canada (being "AHpieCanadaManufacturer") with one of the Hpie Canada suppliers, he has the permission to receive incomplete orders before their due dates (interaction law 4.3). However, he has to pay 15% of the price value as taxes if the deliver is immediately (carry-on) or if the deliver address is in Canada (environment law 1.2).

If the manufacturer decides to buy in Hpie (being "AHpieManufacturer"), he pays just 5% of the price value as taxes if the deliver is immediately (carry-on) or if the deliver address is in the state of the Hpie headquarter (environment law 1.4).

3.3. A Scenario where Laws Are Restricted

To exemplify how the contextual composition of laws can restrict regulations, a scenario is given by the organization laws 2.1 and 2.2, and it is illustrated in the left side of Fig. 6. Hpie Argentina is regulated through the organization law 2.2, but as Hpie is its main organization, it also can be regulated through the organization law 2.1. In this way, Hpie Argentina distributors are not allowed anymore to sell directly to customers (see that the dashed law from the left side of Fig. 6. – "PermissionToDistributors-SaleToCustomers" – is not presented in Hpie).

3.4. A Scenario where Laws Are Relaxed

To exemplify how the contextual composition of laws can relax regulations, a scenario is given by organization laws 2.4 and 2.3, and it is illustrated in the right side of Fig. 6. Dellie Chile is regulated through the organization law 2.4, but as Dellie is its main organization, it also can be regulated through the organization law 2.3. In this way, Dellie Chile suppliers are allowed to sell direct to customers (see that the dashed law from the right side of Fig. 6. – "PermissionToSuppliersSaleToCustomers" – is only presented in Dellie).

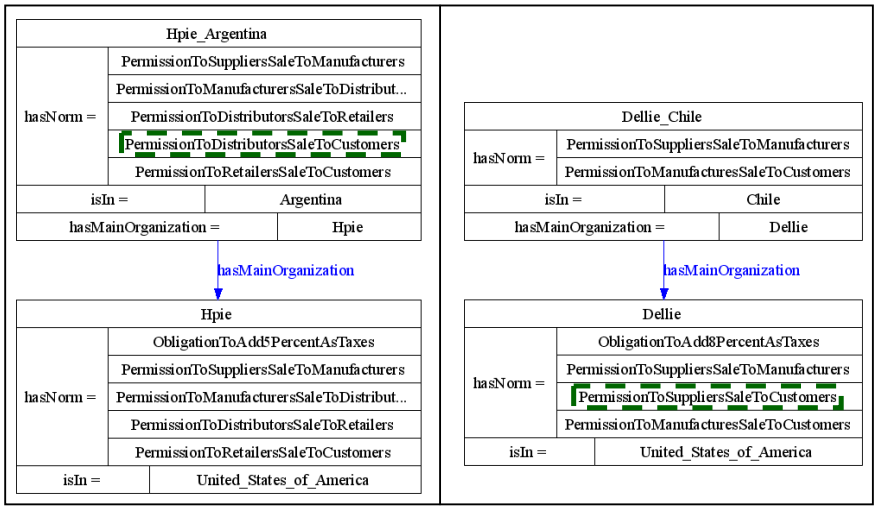

Fig. 6. Laws restricting and relaxing regulations because of contextual compositions of laws

3.5. A Scenario where Laws Are Conflicting

To exemplify how contextual compositions of laws can raise conflicts, a scenario is given by the role laws 3.3 and 3.4, and it is illustrated in Fig. 7. Dellie Brazil suppli-

ers and customers are regulated through laws that state the same subject (deadline to ship orders) in an opposite way. The role law 3.3 states the obligation that suppliers have to ship orders until their due dates, but the role law 3.4 states that customers can receive their orders until one day after their due dates.

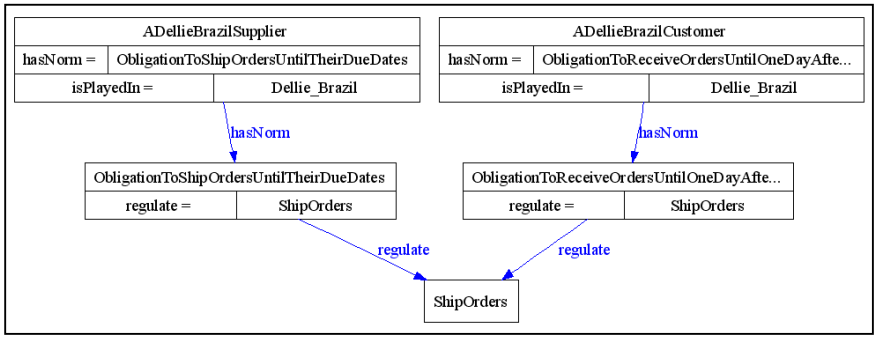

Fig. 7. Laws conflicting in a regulation because of contextual compositions of laws

In this work, we don't make any assumption about the problem of how resolve raised conflicts when laws from the same regulatory context state the same subject in an opposite way. However, we suggest enhance conflicted laws with priorities in order to minimize the problem.

3.6. Implementation

Our case study was implemented inside the Eclipse platform [2], using the Java programming language [8] and the Jena API [12]. The Jena API was used as a programmatic environment for OWL [15] and as a rule based inference engine (rules were written following the Jena rule syntax). The Protégé Editor [17] was used to extend and instantiate the normative meta-ontology. Our agents were implemented in JADE [11], extending its Agent class with both an attribute for their locations and two behaviors. In Fig.8 is illustrated part of our implemented agent code.

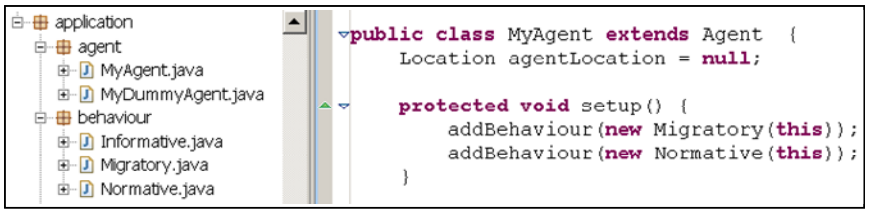

Fig. 8. Part of our implemented agent code

The two implemented behaviors are the ones called Migratory and Normative. The migratory behavior makes agents move randomly from one place to another. The

normative behavior informs agents about the current regulations of contextual laws. Once an agent migrates, its location attribute is updated and, consequently, the contextual laws which he is bound to, at the given moment in time, are also changed.

JADE containers were also used in our implementation to represent environments and organizations, offering possible locations for agents with the migratory behavior, i.e. mobile agents, to go. North America, South America, Canada, United States of America, Argentina, Brazil, Chile, Dellie, Hpie, Hpie Canada, Hpie Argentina, Dellie Brazil and Dellie Chile are the examples of possible locations from our case study.

In Fig. 9 is illustrated part of our system's world with agents inside the United States of America, Canada, Brazil and Chile, all JADE containers. For instance, a mobile agent is in Brazil. Once this agent migrates, its location attribute is updated. The normative behavior always checks this attribute to get the current agent location. For agents, the normative behavior informs the current contextual regulation of the system. For the system, the normative behavior supports the enforcement of laws.

Fig. 9. Part of our system's world implemented in Jade

Our normative behavior is based on rules for contextual law compositions and retrievals. These rules are *ontology-driven*, i.e. they must be created based on our normative meta-ontology concepts and on these concepts' relationships (see Fig. 1.). We implemented our normative behavior in Java and wrote the rules by using the Jena rule based engine syntax [12]. These rules define which set of contextual laws are regulating agent actions at the run-time. Rules can be activated and deactivated, also at run time, in order to change the system current contextual regulations. To activate rules, it is necessary to remove rules' comment marks; to deactivate rules, it is necessary to insert rules' comment marks, both in the set of rules.

All the rules used for the scenarios described in the previous subsections are presented in Table 1. When rule 1 is activated, it states that an environment will also be regulated with its owner environment laws (norms); when rule 2 is activated, it states that an organization will also be regulated with its main organization laws; when rule 3 is activated, it states that an organization will also be regulated with its environment laws; when rules 1, 2 and 3 are activated, they state that an organization will also be regulated with laws from its main organization and environments; when rule 4 is activated, it states that a role will also be regulated with its organization laws. When

rules 1, 2, 3 and 4 are activated, they state that a role will also be regulated with laws from its organization, its organization's main organization and environments.

Table 1. Rules for contextual compositions of law

Rule 1- [ruleForEnvironmentWithOwnerEnvironmentNorm:
(?Environment *belongsTo* ?OwnerEnvironment)
(?OwnerEnvironment *hasNorm* ?OwnerEnvironmentNorm)
-> (?Environment *hasNorm* ?OwnerEnvironmentNorm)]

Rule 2- [ruleForOrganizationWithMainOrganizationNorm:
(?Organization *hasMainOrganization* ?MainOrganization)
(?MainOrganization *hasNorm* ?MainOrganizationNorm)
-> (?Organization *hasNorm* ?MainOrganizationNorm)]

Rule 3- [ruleForOrganizationWithEnvironmentNorm:
(?Organization *isIn* ?Environment)
(?Environment *hasNorm* ?EnvironmentNorm)
-> (?Organization *hasNorm* ?EnvironmentNorm)]

Rule 4- [ruleForRoleWithOrganizationNorm:
(?Role *isPlayedIn* ?Organization)
(?Organization *hasNorm* ?OrganizationNorm)
-> (?Role *hasNorm* ?OrganizationNorm)]

4 Related Work

Our work was compared to OMNI (*Organizational Model for Normative Institutions*) [19]. OMNI is a framework for modeling agent organizations composed of three dimensions: *Normative*, *Organizational* and *Ontological*. OMNI contains the three levels of abstractions with increasing implementation detail: the *Abstract Level*, which has the statutes of the organization to be modeled, the definitions of terms that are generic for any organization and the ontology of the model itself; the *Concrete Level*, which refines the meanings defined in the previous level, in terms of norms and rules, roles, landmarks and concrete ontological concepts; and, finally, the *Implementation Level*, which has the Normative and Organizational dimensions implemented in a given multi-agent architecture with the mechanisms for role enactment and for norm enforcement.

Comparing our work with OMNI, both define a meta-ontology with a taxonomy for regulations in open MAS and use norms to recommend right and wrong behavior. The use of norms can inspire trust in regulated MAS. One difference is that, in OMNI, enforcement is carried out by any internal agents from the system while in our work it can be carried out by some trusted agents or by some system's enforcement mechanisms. A second difference, and the most important, is that in OMNI the idea of regulatory contexts is not explicit and separated in different levels of abstractions, especially for the environment and role law contexts. Our approach is based on the environment, organization, role and interaction regulatory contexts to simplify the enforcement and evolution processes. For instance, the social structure of an organization in OMNI describes, in the same level of abstraction, norms for roles and

groups of roles. Group of roles is used to specify norms that hold for all roles in the group. We use the organization regulatory context to specify organization norms that hold for all roles from an organization and use the role regulatory context to specify role norms, both regulatory contexts from different levels of abstractions.

5 Conclusion

We presented a case study based on a flexible solution to support regulatory dynamics in open multi-agent systems. This approach supports deliberative normative agents with information about the current system enforcement of laws. For future work, we will use the Jess rule engine [7] instead of the Jena one [12], mainly addressing issues like ease-of-use, expressiveness and reasoning.

Acknowledgments

This work was partially funded by the projects ESSMA (CNPq 552068/2002-0) and EMACA (CAPES/COFECUB 482/05 PP 016/04), and by CNPq individual grants.

References

1. Dey, A. Understanding and using context. Personal Ubi /comput 2001; 5(1)
2. Eclipse. Electronic available in: <http://www.eclipse.org/>. Accessed in January 2006.
3. Felicíssimo, Carolina Howard; Lucena, Carlos; Briot, Jean-Pierre; Choren, Ricardo. An Approach for Contextual Regulations in Open MAS. In: Eight International Bi-Conference Workshop on Agent Oriented Information Systems (AOIS-2006) at AAMAS, Hakodate, Japan, 2006.
4. Felicíssimo, C. H.; Lucena, C. J. P. de: An Approach to Regulate Open Multi-Agent Systems Based on a Generic Normative Ontology. First Workshop on Software Engineering for Agent-oriented Systems (SEAS, 2005). Brazil, 2005. Electronic available in: <http://www.les.inf.puc-rio.br/seas2005/file/cFelicissimo.pdf>. Accessed in January 2006.
5. Felicíssimo, C.H.; Lucena, Carlos; Carvalho, G.; Paes, R..: Normative Ontologies to Define Regulations Over Roles in Open Multi-Agent Systems. AAAI Fall Symposium "Roles, an Interdisciplinary Perspective: Ontologies, Programming Languages, and Multiagent Systems", Technical Report FS-05-08. USA. November, 2005. ISBN 978-1-57735-254-9.
6. Ferber, J.; Gutknecht, O.; Michael, F.: From Agents to Organizations: an Organization View of Multi-Agent Systems. In Proc. of the AOSE, 2003. Australia, 2003.
7. Friedman-Hill, E.: Jess, the Rule Engine for the Java Platform. At Sandia National Laboratories. Electronic available in: <http://herzberg.ca.sandia.gov/>. Accessed in January 2006.
8. Gosling, J.; Joy, B.; Junior, G.L.S.; Bracha, G.: The Java Language Specification. ISBN 0-201-31008-2. Electronic available in: <http://java.sun.com/>. Accessed in January 2006.
9. Gruber, T. R.: A translation approach to portable ontology specifications. Knowledge Acquisition, Vol. 5, Issue 2. June, 1993. Pages 199-220. ISSN: 1042-8143.

10. Hewitt, C.: Open Information Systems Semantics for Distributed Artificial Intelligence. Artificial Intelligence, V. 47, Issue 1-3. 1991. p. 79 - 106. ISSN: 0004-3702.
11. JADE. Electronic available in: <http://jade.tilab.com/>. Accessed in January 2006.
12. Jena. Electronic available in: <http://jena.sourceforge.net/>. Accessed in January 2006.
13. Jennings, N. R. On Agent-Based Software Engineering. Artificial Intelligence, 117 (2) pages 277-296. 2000.
14. Jennings, N.; Sycara, K. and Wooldridge, M.: A Roadmap of Agent Research and Development. Journal of Agents and Multi-Agent Systems, 1:7-38, 1998.
15. McGuinness, D. L. and Harmelen, F. v.: OWL Web Ontology Language Overview. Electronic available in: <http://www.w3.org/TR/owl-features/>. Accessed in January 2006.
16. Noy, N; Rector, A. (Eds.): Defining N-ary Relations on the Semantic Web: Use with Individuals. Electronic available in: <http://www.w3.org/TR/swbp-n-aryRelations/>. Accessed in January 2006.
17. Protégé. Electronic available in: <http://protege.stanford.edu/>. Accessed in January 2006.
18. Thomas, G. e Williams, A. B.: Roles in the Context of Multiagent Task Relationships. AAAI Fall Symposium "Roles, an Interdisciplinary Perspective: Ontologies, Programming Languages, and Multiagent Systems", Technical Report FS-05-08. ISBN 978-1-57735-254-9. USA. November, 2005.
19. Vázquez-Salceda, J.; Dignum, V.; Dignum, F.: Organizing Multiagent Systems. Journal of Autonomous Agents and Multi-Agent Systems, 11, 307-360, 2005.
20. Weyns, D.; Parunak, H.V.D.; Michel, F.; Holvoet, T. and Ferber, J.: Environments for Multiagent Systems State-of-the-Art and Research Challenges. In Environments for Multi-Agent Systems (E4MAS, 2004), Springer 2005, p.1-47. ISBN 3-540-24575-8.