An Approach for Contextual Regulations in Open MAS

Carolina Felicíssimo
DI – PUC-Rio
R. M. de S. Vicente 225
Gávea RJ, Brazil
cfelicissimo@inf.puc-rio.br

Carlos Lucena
DI – PUC-Rio
R. M. de S. Vicente 225
Gávea RJ, Brazil
lucena@inf.puc-rio.br

Jean-Pierre Briot
LIP6
8 rue du Capitaine Scott
75015 Paris, France
jean-pierre.briot@lip6.fr

Ricardo Choren
SE/8 – IME
Pça Gen Tibúrcio 80
Urca RJ, Brazil
choren@de9.ime.eb.br

ABSTRACT
In open multi-agent systems, agents can enter and leave the system freely. Seeking to avoid a possible chaotic scenario, we propose to monitor these systems following an approach for the contextual regulation of laws. In our solution, agent actions can be enforced by laws from four regulatory contexts: *Environment*, O*rganization*, *Role* and *Interaction*. The composition of laws from these different levels permits contextual regulations in a more flexible and complete regulation for both the modeling and implementation phases. Our approach defines a top-down contextual modeling of laws, a normative ontology to formalize the regulatory contexts, and an ontology-driven rule support for law retrieval. This approach provides a straightforward method to implement regulations according to the defined regulatory contexts, yielding law evolution, contextual composition of laws, law consistency check and agent law reasoning.

Keywords
Contextual regulation, normative ontology, law-awareness

1. INTRODUCTION
Multi-agent systems (MAS) have emerged as a promising approach to develop information systems that are composed of several goal-oriented problem-solving entities [15]. Openness [12] has led to software systems that have no centralized control and that are formed of autonomous entities, such as agents. MAS can be considered an open system when it presents the following characteristics [9]:

Heterogeneity: agents are possibly developed by different parties, in different languages, with different purposes and preferences.

Accountability: agent actions must be monitored to detect the execution of behaviors that may not be according to the overall expected functioning of the system.

Social change: agent societies are not static; they may evolve over time by altering, eliminating or incorporating rules. Thus, the system requires flexibility to accommodate future changes.

We assume, in this work, 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. This kind of system does not prevent agents from executing actions that violate norms; it penalizes infringing agents for doing so. For instance, agents can be penalized by punishments associated to norms.

In regulated open MAS, agents may be heterogeneous, but they all must know (or be aware of) the actions, norms and penalties of the regulated systems. In this same sense, information must be modeled and expressed in a meaningful and precise way so that developers can create agents that effectively execute in the open MAS and reason about the system's norms and their associated penalties. The 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 actual agent regulation during the system execution. Moreover, this mechanism should be easy to operate so that norms can be created, deleted and modified at run-time.

The purpose of this paper is to try to provide a simple way to model and implement norms for contextual regulations in open MAS. More precisely, we have defined that the laws that will regulate an open MAS should be written according to a top-down modeling, which is limited to four defined regulatory contexts. The laws are formalized into an instance of a normative ontology and are retrieved according to a dynamic contextual composition of ontology-driven rules. This mechanism allows for accountability since the system and the normative ontology-driven agents have the information about which norms should be enforced. Furthermore, the mechanism also allows for social changes since it is able to dynamically define the norms that should be applied to an agent in a given context.

The current implementation of the proposed approach is carried out by instantiating and, probably, extending (with domain concepts) a domain independent normative ontology, according to our top-down modeling of laws. We use the Jena API [13] to deal with ontologies, to define the ontology-driven rules and to make rule-based inferences for law retrieval.

It is important to stress that in this work we do not make any assumptions about how or if an agent decides to comply with norms. With our approach, we allow the modeling of the system norms and sanctions, and the definition of rules that show which norms should be considered in a given context. Thus, we provide the ways for agents to reason about norm compliance while leaving the agents' implementation to the agent developers. Agents that are aware of the system laws are more likely to perform correctly and, thus, achieve their goals faster.

The remainder of this paper is organized as follows. Section 2 presents our approach for contextual regulations in open MAS, including the top-down contextual modeling of laws, the normative ontology to formalize the regulatory contexts and the rule support for law retrieval. Section 3 describes a simple case study that involves different dynamic contextual compositions of laws from two multinational companies. Section 4 briefly

discusses some related works. Finally, Section 5 offers our conclusions and outlines some future work.

2. CONTEXTUAL REGULATIONS IN OPEN MULTI-AGENT SYSTEMS

A multi-agent system is constituted, mainly, by environments, organizations, agents, agent roles and agent interactions [14]. Environments [24] are discrete computational locations (similar to places in the physical world) that provide conditions for agents to inhabit it. For environments there can be refinement levels, such as a specialization relationship (e.g. country, state, city, street, etc.), but there cannot be overlaps (e.g. there cannot be two countries in the same place). Moreover, an environment can have many organizations inside it, i.e. partitions and groups of entities such as departments, communities and societies [8]. An organization defines its roles and, probably, can have sub-organizations too.

An organization is composed of a group of agents playing roles inside it in order to achieve their goals. An agent can be in many organizations, but each organization must belong to just only one environment [19]. Yet, agents with the mobility characteristic can move from one environment to another or can register or leave organizations, assuming different roles and obeying or not all their defined laws.

Norms can control environments, organizations, agent roles and agent interactions by monitoring the actions performed by agents in the regulated open MAS. Norms can also define which actions are *permitted*, *obliged* and *prohibited* to be executed. A permitted norm defines that an act is allowed to be performed; an obligated norm defines that an act must be performed; and a prohibited norm defines that an act must not be performed. These three types of norms described represent the three fundamental deontic statuses of an act [1] from Deontic Logic [25]. Deontic Logic makes it possible to address the issue of explicitly and formally defining norms and dealing with the possibility of violation.

To provide norm regulations according to Deontic Logic and law-awareness by agents in open MAS, regulatory mechanisms are desired. However, these mechanisms ought to be easy, simple and flexible enough to be used and to allow law enforcement and evolution processes.

2.1 A Top-Down Contextual Modeling of Laws

In order to help at structuring regulations in open MAS, we defined a top-down contextual modeling of laws, which is an evolution of the ideas presented in [5, 6, 7]. In this work, we propose four regulatory contexts of laws: *Environment Laws*, *Organization Laws*, *Role Laws* and *Interaction Laws*.

Environment Laws are those that are applied to all agents from the regulated environment, independent of its organizations, roles and interactions. *Organization Laws* are those that are applied to all agents from the regulated organization, independent of its roles and interactions. *Role Laws* are those that are applied to all agents playing the regulated role, independent of its interactions. *Interaction Laws* are those that are applied to all agents involved in the regulated interaction. Figure 1 illustrates the boundaries of our four defined regulatory contexts.

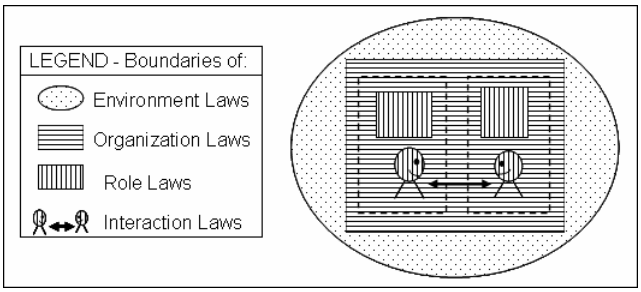

Figure 1. Boundaries of our defined regulatory contexts

In order to enable agents to process the content of the regulatory contexts, these contexts should have their semantic expressed in a meaningful way. Ontologies, i.e. conceptual models that embody shared conceptualizations of a given domain [11], can provide the desired semantic support, making the represented information of a domain easier for agents to automatically process their meanings [16].

2.2 A Normative Ontology to Formalize the Regulatory Contexts

In regulated systems, agents need to be normative entities, i.e. they must be able to take into account the existence of social norms in their decisions (either to follow or to violate a norm) and to react to norm violations by other agents [2]. Normative ontologies, those that have the norm concept as a central asset, supply well-defined information for norm-aware agents to guide their behaviors.

A domain independent normative ontology was constructed to assist contextual regulations in open MAS. This ontology restricts, with norms and their associated penalties, agent actions with instances of environment, organization, role and interaction laws. The first three regulatory contexts (environment, organization and role laws) are represented in the ontology by the six following related main concepts, all at the same hierarchical level: *Environment*, *Organization*, *Role*, *Norm*, *Penalty* and *Action* (see Figure 2).

The six related domain independent concepts defined have specific data associated with them. The *Environment* concept encompasses its norms. The *Organization* concept encompasses its environment and norms. The *Role* concept encompasses the organization where it can be played in and its norms. The *Norm* concept encompasses its associated penalties and regulated actions. The *Penalty* concept encompasses the fine to be levied if its associated norm is violated. The *Action* concept encompasses the actions that must be regulated. The Norm and Penalty concepts are specialized into sub-concepts according to the permitted, obliged and prohibited statuses of an act from Deontic Logic.

Interaction laws are domain dependent and thus they are not present in the normative ontology since the latter is domain independent. Interaction laws must be implemented during the ontology instantiation and extension processes by following the representation pattern from the *Semantic Web Best Practices* document [17]. 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. For instance, suppose that a supplier deals with a customer and the interaction is regulated by a law describing an obligation to pay when the deal is done. The interaction law is represented by a new obligation concept, called for example "*ObligationToPay*", linking the supplier and customer roles.

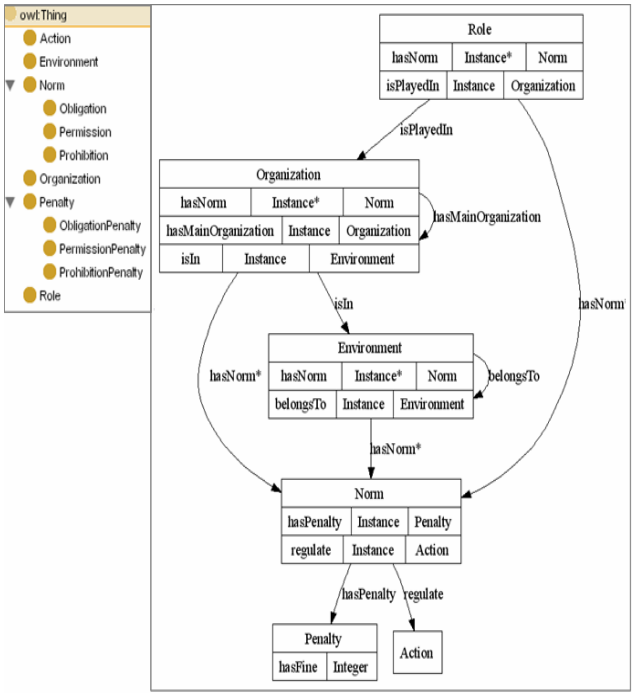

Figure 2. A domain-independent normative ontology

2.3 A Rule Support for Law Retrieval

After classifying all the desired laws for contextual regulations according to our top-down modeling of laws and formalizing them into an instance (possibly extended) of the normative ontology, this ontology instance has to be added to the open MAS project that will be regulated. This instantiated ontology should be complemented by a set of rules, which should be responsible for composing laws from different regulatory contexts and to inform 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 that deals with law compositions according to different regulation contexts is to free the system and the agents from code related to law retrieval. Therefore, for law retrieval, a rule inference engine automatically do both: reads the normative ontology instance (in which data is expressed) and, based on the set of rules, compose the laws according to the contextual regulations expressed there. No implementation code is needed to compose the regulatory contexts or to check out the ontology for law retrieval. This makes our solution more flexible since law management is conducted outside the implementation code. We also offer an implementation of our normative behavior to be added in agents implemented by using JADE [23].

In our approach, defined rules are *ontology-driven*, i.e. they must be created based on the ontology concepts and on these concepts' relationships. Table 1 presents possible rules for contextual compositions of laws, according to our domain independent normative ontology. Rules can be created by linking different types of concepts, e.g. rules 1, 2 and 3, or by linking concepts from the same type, e.g. rules 4 and 5. Furthermore, rules can be created linking concepts indirectly related, e.g. rule 2.

Table 1. Rules for contextual compositions of laws

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

2- [ruleForRoleWithEnvironmentNorm:
(?Role *isPlayedIn* ?Organization)
(?Organization *isIn* ?Environment)
(?Environment *hasNorm* ?EnvironmentNorm)
-> (?Role *hasNorm* ?EnvironmentNorm)]

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

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

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

Rule 1 is an example of a rule linking different types of concepts (the Role and the Organization concepts). This rule expresses how a role can also be regulated through its organization norms. Since a role is played in an organization and the Role and Organization concepts are linked by the "*isPlayedIn*" relationship (see Figure 2), the role's organization instance is discovered by following this path (second line of rule 1). As the Organization and Norm concepts are linked by the "*hasNorm*" relationship (again, see Figure 2), the organization norm instances are discovered by following this path (third line of rule 1). With the organization norms, the contextual regulation is achieved by the composition of the role norms with the organization norms (fourth line of rule 1).

Rule 4 is an example of rule linking concepts from the same type (Organization concepts). This rule expresses how an organization can also be regulated through its main organization norms. Since an organization and its main organization are linked by the "*hasMainOrganization*" relationship (see Figure 2), the organization's main organization instance is discovered by following this path (second line of rule 4). As the Organization and Norm concepts are linked by the "*hasNorm*" relationship, the main organization norm instances are discovered by following this path (third line of rule 4). With the main organizations norms, the contextual regulation is achieved by the composition of the organization norms with its main organizations norms (fourth line of rule 4).

Rule 2 is an example of rule linking two concepts indirectly related (the Role and Environment concepts). This rule expresses how a role can also be regulated through its environment norms. Since a role is played in an organization and this organization belongs to an environment, the role's environment instance is

discovered by following the "*isPlayedIn*" relationship from the role to its organization (second line of rule 2) and, than, following the "*isIn*" relationship from its organization to its environment (third line of rule 2). As the Environment and Norm concepts are linked by the "*hasNorm*" relationship (see Figure 2), the environment norm instances are discovered by following this path (fourth line of rule 2). With the environment norms, the contextual regulation is achieved by the composition of the role norms with its environments norms (fifth line of rule 2).

It is important to remark here that, despite the organization concept is the link from the role concept to the environment concept, when a role is regulated through its environment norms too, organization norms can or can not be composed to the norms.

Composition of laws can raise conflicts, restrict or relax regulations. All cases happen when laws address the same subject in an opposite way. Thus, the fundamental deontic statuses of an act are violated, e.g., permitted or obliged acts cannot be prohibited, obliged acts must be permitted too, etc. Conflicts are raised when opposite laws are from the same level of regulation, e.g. an act is permitted and prohibited in different environment laws. Restrictions happen when composed laws from a more abstract level are more restrictive, e.g. an organization law is more restrictive than a role law. Relaxations happen when composed laws from a more abstract level are more relaxed, e.g. an organization law is more relaxed than a role law. Some strategies can be adopted by the regulated system in order to resolve conflicts or to avoid restrictions or relaxations, i.e. to support enforcement consistency. Enhancing conflicting laws with priorities is an example of strategy.

3. CASE STUDY

As an example scenario for our case study, we assume that Dellie and Hpie represent two multinational companies from the real world. Based on corporate Web sites, we created laws for Dellie and Hpie specifically to present our case study. The laws were classified according to our regulatory contexts: *Environment*, *Organization*, *Role* and *Interaction*.

To simplify the presentation, our system's world consists of eight environments (World, 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). Figure 3 illustrates the world designed to present our case study.

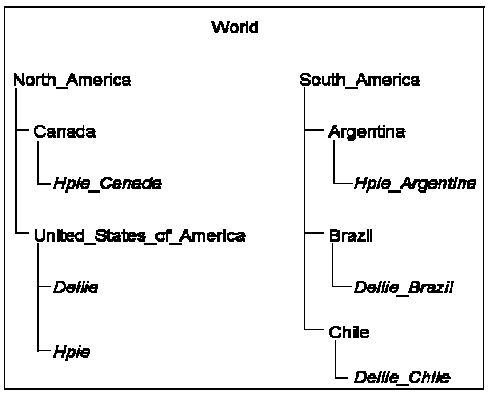

Figure 3. Our defined world

The laws we created were classified according to the four proposed regulatory contexts.

1. **Examples of Environment Laws:**

   1.1. In the world, every organization has to honor its responsibilities (pay salary to all of its employees, etc.)

   1.2. In North America, every organization has to prolong its product life span.

   1.3. In the United States of America, everybody has to conserve energy consumption.

   1.4. In South America, everybody has to minimize the stress on the environment while alive.

   1.5. In Brazil, every organization has to reduce or eliminate materials for disposal.

2. **Examples of Organization Laws:**

   2.1. In the world, every Hpie organization, independent of all of its other roles, 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 Dellie Chile, independent of all of its other roles, the sales of the organization's products can just be made between: suppliers and manufacturers, or manufacturers and customers.

   2.3. In Dellie Brazil, independent of all other its roles, just suppliers and manufacturers are permitted to sell the organization's products to customers.

3. **Examples of Role Laws:**

   3.1. In Dellie, independent of all others roles that customers deal with, only complete orders can be shipped.

   3.2. In Hpie Canada, independent of all others roles that suppliers deal with, orders are always shipped on their due dates.

   3.3. In Hpie Argentina, independent of all others roles that customers deal with, a down payment of 10% on orders must be paid.

   3.4. Customers must pay for every shipped order.

4. **Examples of Interaction Laws:**

   4.1. Suppliers must answer all manufacturers' RFQs (Request for Quotes).

   4.2. Manufacturers can only send five RFQs/day to each supplier for each of their products offered.

   4.3. In Dellie Brazil, a manufacturer cannot sell to another manufacturer.

We classified the presented laws according to the definitions of the regulatory contexts given in Section 2.1, but it is known that the boundary between each context, sometimes, is unclear. For these cases, the classifications can be discussed.

An example of discussion can be given with the classification of rule 3.1. This rule was classified as a Role Law instead of an Organization Law, because it is a law for the customer role. This law is independent of all other organization roles, e.g. suppliers still can ship incomplete orders to manufacturers. If the law states

that in the organization only complete orders can be shipped (without any distinction of roles), then the law should be classified as an Organization Law, because it will influence all organization roles.

Another example of discussion is the classification of rule 4.3. This rule was classified as an Interaction Law also instead of an Organization Law, because it is a law for the interaction between manufacturers in Dellie Brazil. This law is independent of all other organization roles, e.g. suppliers still can buy from one Dellie Brazil manufacturer and sell to another, characterizing an indirect sale between the manufacturers.

3.1 Ontology Extension

The normative ontology can be extended for specific domains in order to formalize domain related concepts and to represent interaction laws. In this case study, because the supplier, manufacturer, distributor, retailer and customer roles have distinct goals and, consequently, execute specific plans and actions to achieve them, they were formalized extending the Role concept from the normative ontology.

Interaction laws 4.1, 4.2 and 4.3 also extend the normative ontology by following the representation pattern presented in Section 2.2, i.e. the relation object itself is created as a Norm sub-concept and it links Role sub-concepts. For instance, law 4.1 (illustrated in Figure 4) is represented by the new Norm sub-concept called "*ObligationToAnswerAllRFQs*" linking the "*Supplier*" and "*Manufacturer*" Role sub-concepts.

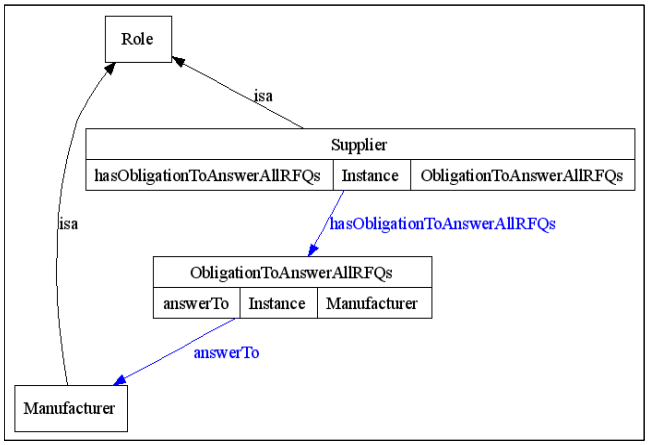

Figure 4. An example of an interaction law's representation

3.2 Ontology Instantiation

The normative ontology has to be instantiated for contextual regulations in open MAS. This ontology should have data assigned to, based on the top-down modeling of laws. More precisely, to represent all the described laws in the ontology instance, we must create: all environments from any law as instances of the Environment concept; all organizations from any law as instances of the Organization concept; all roles from any law as instances of the Role concept; all interactions from any law as instances of new Norm and Role sub-concepts; all actions, norms and penalties as instances of their associated concepts.

For example, Figure 5 illustrates the instances (connected to their associated concepts by links labeled as *io* – instance of) created for the interaction law 4.1. This law is represented by the "*ObligationToAnswerAllRFQsInHpie*" norm instance linking the role instances "*AHpieSupplier*" and "*AHpieManufacturer*".

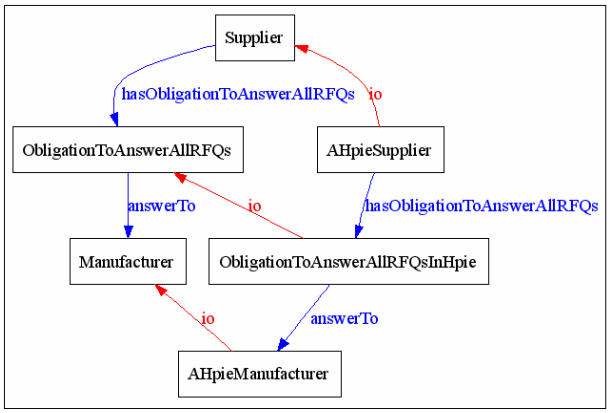

Figure 5. An example of an interaction law's instance

Another example of an ontology instantiation is illustrated in Figure 6. The role law 3.2 is represented by the "*AHpieCanadaSupplier*" role instance, which in turn is regulated by the "*ObligationToShipOrdersInTheirDueDates*" norm instance. The role is played in the "*Hpie_Canada*" organization instance. The norm regulates the "*ShipOrders*" action instance and has the "*ObligationPenaltyToShipOrdersInTheirDueDates*" penalty instance.

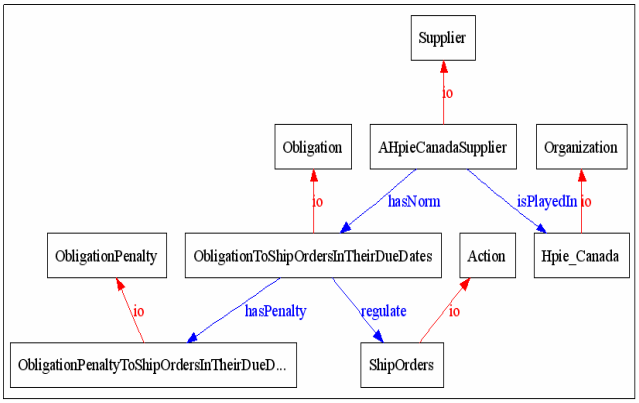

Figure 6. An example of a role law's instance

3.3 Implementation

Our case study was implemented inside the Eclipse platform [4], using the Java programming language [10] and the Jena API [13]. The latter is a programmatic environment for OWL [16] and a rule-based engine. The Protégé Editor [22] was used to extend and instantiate the normative ontology. The agents from our case study were implemented using JADE [23], in which the normative behavior for contextual regulations from our approach was also implemented.

Our implementation can be summarized in: classifying the desired laws according to our top-down contextual modeling, instantiating and extending the normative ontology and, finally, adding in agents both a Location attribute (agents' locations reflect their contexts) and the normative behavior from our approach.

In Figure 7 is illustrated part of our agent code. Our agent extends the Jade agent and has both an attribute for its location and two added behaviors. The migratory behavior is just to make agents

move randomly from one location to another. The normative behavior adapts the agent to the current system contextual regulation of laws.

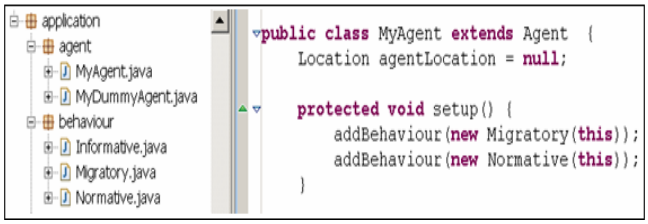

Figure 7. Part of our implemented agent code

The environments, main organizations and organizations from our World are represented by Jade containers (see some environments in Figure 8) and offer possible locations for agents with the migratory behavior (mobile agents) to go. Once an agent migrates, its location attribute is updated. The normative behavior always checks this attribute in order to get the current agent location. For agents, the normative behavior informs the current contextual regulations of the system. For the system, the normative behavior supports the enforcement task.

Figure 8. Part of our world implemented in Jade

Our normative behavior is based on ontology-driven rules for law retrieval. We implemented this behavior in Java and wrote the rules by using the Jena rule based engine syntax [13]. The written rules were presented in Table 1 from this work.

Activations and deactivations of rules define which sets of contextual laws are regulating agent actions at the run-time. Rules can be activated and deactivated, manually, also at run-time, in order to change the system current contextual regulations. To activate rules, it is necessary to remove the rules' comment marks; to deactivate rules, it is necessary to insert the rules' comment marks, both in the set of rules.

A scenario with different possibilities of contextual composition of laws is illustrated in Figure 9. Dellie Brazil will be our focus in the explanation of how regulatory contextual compositions influences law retrieval. Dellie Brazil is located in Brazil. Brazil is located in South America, which in turn is located in the World. Dellie Brazil has Dellie as its main Organization. Dellie is located in United States of America. United States of America is located in North America, which in turn is located in the World. All Organizations and Environments are linked in our normative ontology. The "*hasMainOrganization*" relationship is the one that links an Organization with its main Organization; the "*isIn*" relationship is the one that links an Organization with its Environment; the "*belongsTo*" relationship is the one that links two hierarchical Environments.

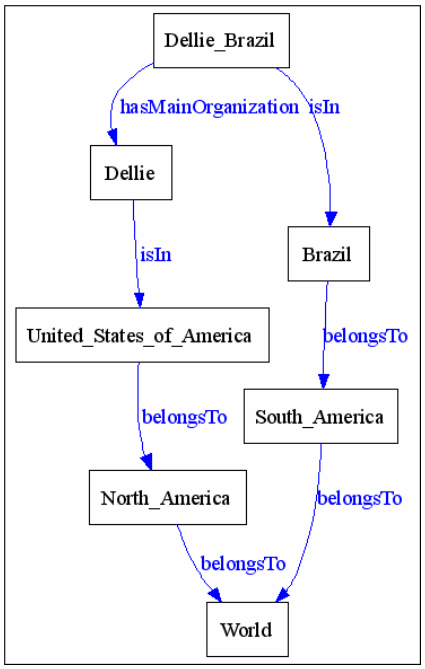

Figure 9. Possibilities for contextual composition of laws

Dellie Brazil is regulated, by default, by laws 2.3 and 4.3. However, the World law 1.1 can be composed to regulate Dellie Brazil too. To do that, for example, rules 3 and 5 from Table 1 have to manually be activated. Doing that, norms from Brazil and South America will also be composed to the Dellie Brazil norms because they are one of the links from Dellie Brazil to the World. If this is not desired (for instance, just the World norms are wanted for the contextual composition of laws), a specific rule can be written as follows (see Figure 10):

```
6- [ruleForOrganizationWithWorldNorm:
      (World hasNorm ?WorldNorm)
        -> (?Organization hasNorm ?WorldNorm) ]
```

Figure 10. An organization regulated with also World norms

Dellie Brazil can also be regulated by norms from Dellie (activating rule 4), United States of America, North America and/or World (activating rules 3 and 5).

Composition of laws can relax or restrict contextual regulations. In our example, composing Dellie Brazil with all laws from Brazil, South America, Dellie, United States of America, North America and World constrained the regulation in different contexts. For example, the composition with the Dellie law 3.1 limited Dellie Brazil suppliers and manufacturers to ship only complete orders.

Another scenario for contextual regulations of laws can be given by composing the organization law 2.1 with the role law 3.2. The result is that in Hpie Canada suppliers can only ship manufactures' orders on their due dates and cannot sell to others Hpie roles.

Our approach of identifying, classifying and implementing laws according to regulatory contexts is extremely important to manage regulated dynamic systems. In the lifecycle of this kind of systems, new laws are included and existing ones are modified

to support new practices and new regulations. Consequently, it is important to have a flexible design that can incorporate these changes easily. For instance, in order to make a negotiation process easy, interaction and role laws can evolve to be more flexible, whereas environment and organizations laws do not need to evolve too.

4. RELATED WORK

Our approach for contextual regulations in open MAS is compared with XMLaw [18]. XMLaw encompasses a declarative language and a software implementation. The language supports a conceptual model for developing laws in open MAS. This model is composed of static and dynamic definitions. The implementation is to allow the enforcement of laws through the interception of agents' interaction. In XMLaw, regulations take place at the level of interaction laws by managing agents' interactions in order to achieve higher degrees of predictability.

Comparing our work with XMLaw, three main differences can be assessed. The first one addresses the defined regulatory contexts. In XMLaw, just interaction laws are defined and regulations are based only on this level. In our work, interaction laws can be also composed with environment, organization and role laws for a more complete regulation. The second main difference between the works is how the enforcement is carried out when agents do not act according to the defined laws. In XMLaw, enforcement is carried out *a priori*, i.e. it intercepts messages and checks them to avoid law violations. In our work, enforcement is carried out *a posteriori*, i.e. laws are checked and if there was a law violation, its associated penalty (punishment) is assigned to the infringing agent. Thus in our approach, the privacy of messages exchanged between agents is maintained and the overload consequence of the interception process of all changed messages does not exist. Finally, the third main difference is that in our work the regulated system is simulated, i.e. all the regulated parts have to be known a priori, whereas in XMLaw, as it enforces law only in the interaction level, this is not necessary.

Furthermore, our work was compared to OMNI [3]. OMNI (Orgazitional Model for Normative Institutions) 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 the 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 agents or not. A second difference, and the most important, is that in OMNI the idea of regulatory contexts is not explicit, 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 (or sub-organizations) 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 in this work an approach that is able to: implement regulations according to defined regulatory contexts; manage law evolutions; retrieve laws, use dynamic contextual composition; automatically check the consistency of pre-defined laws; and support *black-boxes* agents to reason about laws.

Contexts provide a modular strategy to implement laws according to different regulatory levels. This modular strategy permits flexibility for a dynamic composition of laws from the same or different contexts; these are independently related.

Our approach provides a simple way to manage law evolutions in two different cases. The first case is when laws need to be added, updated or deleted to the regulated system. For this case, simply updating both 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 check it at run-time. The second case for law evolution is when the contextual composition of laws needs 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 the agents are checking the set of rules whenever they retrieve every law.

Our solution encompasses a top-down contextual modeling of laws, a normative ontology to formalize the regulatory contexts and an ontology-driven rule support for law retrieval. This solution provides better data retrieval because it is based on a set of rules plus a rule inference engine instead of based on various locations inside the implemented codes. The set of rules specifies, according to the top-down contextual modeling, the way the ontology should be searched to obtain its data (the instantiated laws). As this set can be changed, at run-time, according to the normative ontology and new desired contextual regulations, it allows a dynamic contextual composition of laws to retrieve data.

The rules and the rule inference engine also allow an automatic consistency check for pre-defined laws. For this, as an example, during the enforcement consistency check, rules have to address points where conflicts can occur (e.g. by the composition of laws) and offer a solution to resolve them. An example of resolution is given by enhancing rules with priorities.

Finally, agent reasoning about laws is supported by our normative ontology that offers a well-defined enforcement model for ontology-driven agents. Furthermore, these agents can also be treated as *black-boxes* entities when they assume a normative behavior. In this case, no extra implementation inside the agents' code is necessary to awaken them about the system's contextual regulations.

Our approach for contextual regulations in open multi-agent systems can also be used for different contexts. To do that, desired regulatory contexts can be created as new model levels in the top-down modeling of laws. Thus, they must be semantically described in the normative ontology as new related concepts and, finally, they will be able to participate in the new regulatory contextual compositions for law retrieval. As a future work, this possibility will be better studied in order to, probably, recognize an implicit process for contextual regulations. For that, we know that new case studies from different domains and complexities have to be conducted.

6. 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.

7. REFERENCES

[1] Alberti, M.; Gavanelli, M.; Lamma, E.; Mello, P.; Torroni, P. and Sartor, G.: *Mapping Deontic Operators to Abductive Expectations*. In Proc. of *NorMAS 2005*. UK.

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

[3] Dignum, V.; Vázquez-Salceda, J.; Dignum, F.: *A Model of Almost Everything: Norms, Structure and Ontologies in Agent Organizations*. In Proc. of the *AAMAS, 2004*. ISBN 1-58113-864-4. Pages 1498-1499.

[4] Eclipse. Version 3.0. Electronic available in <http://www.eclipse.org/>. Accessed in January, 2006.

[5] Felicíssimo, C.; Sardinha, C. and Lucena. C. *Sweetening Regulated Open Multi-Agent Systems with Support for Agents to Reason about Laws*. MCC 32/05. Technical report. PUC-Rio. Brazil. 2005. ISSN 0103-9741.

[6] 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.

[7] 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.

[8] Ferber, J.; Gutknecht, O.; Michael, F.: *From Agents to Organizations: an Organization View of Multi-Agent Systems*. In Proc. of the *Workshop on Agent-Oriented Software Engineering* (*AOSE, 2003*). Australia, 2003

[9] Garcia-Camino, A.; Noriega, P.; Rodríguez-Aguillar, J.A. Implementing Norms in Electronic Institutions. In Proc. of the 4th Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems, volume 2:667-673, 2005.

[10] Gosling, J.; Joy, B.; Junior, G.L.S.; Bracha, G.: *Java*. ISBN 0-201-31008-2. Version 2.0. Electronic available in <http://java.sun.com/docs/books/jls/download/langspec-2.0.pdf>. Accessed in January 2006.

[11] Gruber, T. R.: A translation approach to portable ontology specifications. *Knowledge Acquisition*, Vol. 5, Issue 2. June, 1993. Pages 199-220. ISSN: 1042-8143.

[12] Hewitt, C.: *Open Information Systems Semantics for Distributed Artificial Intelligence*. Artificial Intelligence. Volume 47, Issue 1-3. 1991. Pages: 79 – 106. Elsevier Science Publishers Ltd. Essex, UK. ISSN: 0004-3702.

[13] Hewlett-Packard Development Company: *Jena – a Semantic Web Framework for Java*. Version 2.3. Electronic available in <http://jena.sourceforge.net/>. Accessed in January 2006.

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

[15] Jennings, N.; Sycara, K. and Wooldridge, M. A Roadmap of Agent Reesearch and Development. Journal of Agents and Multi-Agent Systems, 1:7-38, 1998.

[16] McGuinness, D. L. and Harmelen, F. v.: *OWL Web Ontology Language Overview*. In: <http://www.w3.org/TR/owl-features/>.

[17] Noy, N; Rector, A. (Eds.). Semantic Web Best Practices and Deployment Working Group: *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.

[18] Paes, R. de B.; Carvalho, G. R. de; Lucena, C.J.P. de; Alencar, P.S.C.; Almeida, H.O. de; Silva, V. T. da: Specifying Laws in Open Multi-Agent Systems. In: *Agents, Norms and Institutions for Regulated Multiagent Systems* (*ANIREM*), 2005, Utrecht, The Netherlands.

[19] Silva, V. T. da. *From a conceptual framework for agents and objects to a multi-agent system modeling language*. Ph.D. Thesis. Port. 252 p. PUC-Rio. March 2004.

[20] Smith, M. K.; Welty, C. and McGuinness, D. L.: *OWL Web Ontology Language Guide.* Electronic available in <http://www.w3.org/TR/owl-guide/>. Accessed in January 2006.

[21] Sommerville, I. *Software Engineering*, 7th Edition. Addison-Wesley. ISBN 0-201-17568-1.

[22] Stanford University School of Medicine: *Protégé - A free, open source ontology editor and knowledge-base framework*. Electronic available in <http://protege.stanford.edu/>. Accessed in January, 2006.

[23] Tilab Co. *JADE - Java Agent DEvelopment Framework*. In: <http://jade.tilab.com/>

[24] 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*), USA. Lecture Notes in CS 3374 Springer 2005, p.1-47. ISBN 3-540-24575-8.

[25] Wright, G. H. von: *Deontic Logic*. Mind, New Series, Vol. 60, No. 237 (January, 1951), pages. 1-15.