Regulating Open Multi-Agent Systems with DynaCROM

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

[1]Dept. de Info. – PUC-Rio, R. M. de S. Vicente 225, 22.453-900, Rio de Janeiro, Brazil
{cfelicissimo, lucena}@inf.puc-rio.br

[2]Laboratoire d'informatique de Paris 6 – 8, rue du Capitaine Scott, 75015 Paris, France
Jean-Pierre.Briot@lip6.fr

[3]SE/8 – IME, Praça Gen Tibúrcio 80, Urca, Rio de Janeiro, RJ, 22290-270, Brazil
choren@de9.ime.eb.br

Abstract. *This paper presents our ongoing work for dynamic contextual regulations in open multi-agent systems – called DynaCROM. DynaCROM is a straightforward method to smoothly apply and manage regulations in open multi-agent systems, enforcing customized compositions of contextual laws. DynaCROM is based on a top-down modeling of contextual laws, on a normative meta-ontology for laws semantics and on a rule mechanism for composing contextual laws.*

1. Introduction

Agent-based computing is rapidly emerging as a powerful technology for the development of distributed and complex software systems [Jennings et al. 1998]. This type of systems can be composed of several agents, executing as goal-oriented problem-solving entities with autonomy, mobility, adaptability, etc. [OMG 1999]. In open MAS, agents can unrestrainedly leave their MAS and migrate to others, seeking to obtain resources or services not found locally. However, open systems are always subject to unanticipated interactions [Hewitt, 1991] caused by *buggy* or *malicious* entities that do not conform to recommendations of correct behavior. This risk imposes the need for regulatory mechanisms to prevent unreliable actions and to inspire trust for the agents of open MAS.

In open domains, no centralized control is feasible; the key characteristics of such domains are: agent heterogeneity, conflicting individual goals and limited trust [Artikis 2002]. Heterogeneity and autonomy rule out any assumption concerning the way agents are constructed and behave. Thus, an external control, not hard coded inside agent implementations and which can has its data (e.g. laws) dynamically updated, is the only viable solution for regulations in open MAS [Grizard et al. 2006].

This article focuses on the implementation level of a solution for regulations in open MAS, necessary to enable agent societies. This solution follows our approach for dynamic contextual regulations in open MAS [Felicissimo et al. 2006] – now called DynaCROM. DynaCROM is based on a top-down modeling of contextual laws, on a normative meta-ontology for laws semantics and on a rule mechanism for composing contextual laws. The remainder of this paper is organized as follows: Section 2 presents DynaCROM; Section 3 applies DynaCROM in an open MAS; Section 4 compares DynaCROM with two related works; and, finally, Section 5 concludes the work.

2. Regulations in Open MAS

MAS are generally constituted by environments, organizations and agents [Jennings 2000]. Agents play roles and interact for effectively achieving their designed goals. Environments [Weyns et al. 2005] are discrete computational locations (similar to places in the physical world) that provide conditions for agents to inhabit it. Organizations [Ferber et al. 2003] are social locations where groups of agents play roles, seeking to achieve their goals. Agent roles are abstractions that define a set of related tasks [Thomas and Williams 2005]. Agents interact with other agents, from the same or different organizations and environments.

2.1. Contextual Regulations in Open MAS

Environments, organizations, roles and agent interactions suggest different contexts for regulations in open MAS. Contexts are implicit situational information [Dey 2001] that might be used to characterize situations of participants. Modular context refinements allow a more flexible system and provide a better support for developers, while they are maintaining and evolving information.

Context-aware systems use contexts to provide relevant information and/or services to their users, where relevancy depends on the users' tasks [Dey 2001]. In our definition, *regulated context-aware systems* use *laws* for providing information to users about the current system regulation. However, developers of regulated context-aware systems cannot assume that the system laws will be incorporated inside agents because the control over their codes is not public. Laws should be encapsulated in an external solution, which will not affect or interfere in agents' original implementation.

In [Felicíssimo et al. 2006] we proposed an approach for dynamic contextual regulation provisions in open MAS. This approach – now called DynaCROM – represents a regulatory solution implemented as an agent behavior, which continuously informs current system regulations. Agents, aware of this consequence, can add or not this behavior. The main advantages of DynaCROM for developers of regulations in open MAS are: elucidate and consolidate user defined laws according to a top-down modeling; represent modeled laws in a meaningful way (i.e., with a common understanding) for agents; and easily update the system regulation by both evolving its laws in a unique resource and customizing different compositions of contextual laws.

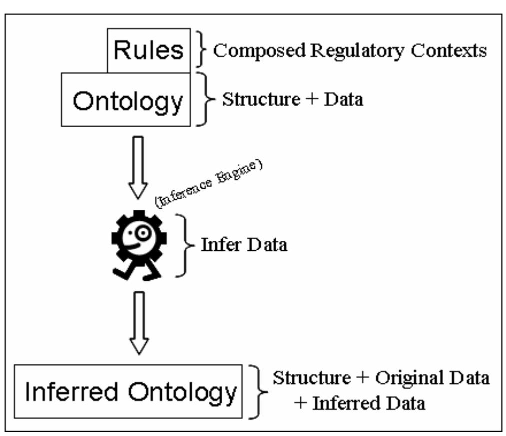

Figure 1. An overview of the DynaCROM approach

Figure 1. illustrates an overview of the DynaCROM approach. There, the ontology structure explicitly represents regulatory contexts (by related concepts) and the ontology data explicitly represents environment, organization, role and interaction laws (by concept instances). Dynamic activations and deactivations of rules are used to manually define customized compositions of contextual laws. A rule-based inference engine automatically deduces composed contextual laws according to the ontology and active rules, and expresses those laws into an inferred ontology.

2.2. A Top-Down Modeling of Contextual Laws

Contexts are tacitly understood by most people, but they are generally hard to elucidate. We believe that classifying contextual laws according to a top-down modeling, it facilitates the developer tasks of elucidation and structuring information. However, we agree that sometimes it is difficult to classify laws according to existing contexts due their subjectivity.

Researches in context-aware applications suggest top-down architectures for contextual modeling [Khedr and Karmouch 1995]. Thus, we addressed in DynaCROM four regulatory modeling contexts: the Environment, Organization, Role and Interaction Laws. *Environment Laws* are applied to all agents in a regulated environment; *Organization Laws* are applied to all agents in a regulated organization; *Role Laws* are applied to all agents playing a regulated role; and *Interaction Laws* are applied to all agents involved in a regulated interaction. The boundaries of environment, organization, role and interaction laws are illustrated in Figure 2. There, for instance, one agent from the environment on the right side interacts with an agent from the environment on the left side, both regulated through an interaction law. Moreover, these agents can also be regulated through customized compositions of their respective environment, organization and/or role laws, for a more precise regulation.

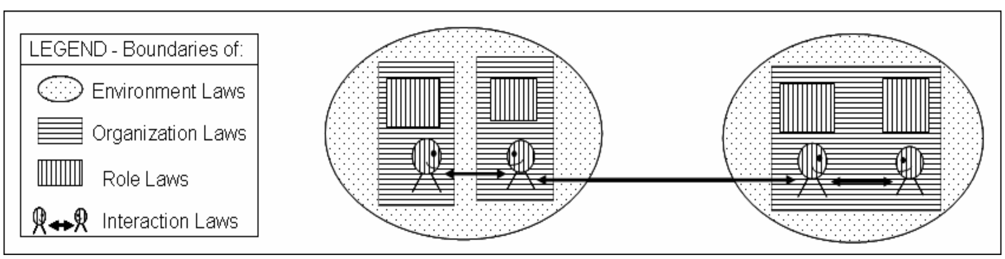

Figure 2. Interactions possibly regulated through compositions of contextual laws

DynaCROM regulatory contexts have different levels of abstraction. Laws from different regulatory contexts can be dynamically composed, restricting or relaxing the current system regulation. Besides, DynaCROM regulatory contexts are not targeted to a particular application domain; they rather represent a minimum set for a general contextual regulation. For more complex MAS, this set should be improved with additions and refinements of particular domain regulatory contexts.

2.3.A Contextual Normative Meta-Ontology

For regulations in open MAS, laws can be represented by sets of norms. Norms should control environments, organizations, roles and agent interactions, defining which actions are *permitted* (allowed to be performed), *obliged* (must be performed), and *prohibited* (must not be performed). Thus, norms, in some way, influence or constrain deliberative normative agents [Castelfranchi et al. 1999] that use norm information to better adapt their behaviors, according to current system regulations.

Norms can be explicitly represented by ontologies. We consider *ontologies* as conceptual models that embody shared conceptualizations of a given domain [Gruber 1993]; *contextual ontologies* as ontologies that keep their contents local (therefore, not shared with other ontologies) [Bouquet et al. 2003]; and, *contextual normative ontologies* as ontologies that keep contextual norm information. Following these considerations, a contextual normative meta-ontology was created for DynaCROM. This ontology is illustrated in Figure 3.

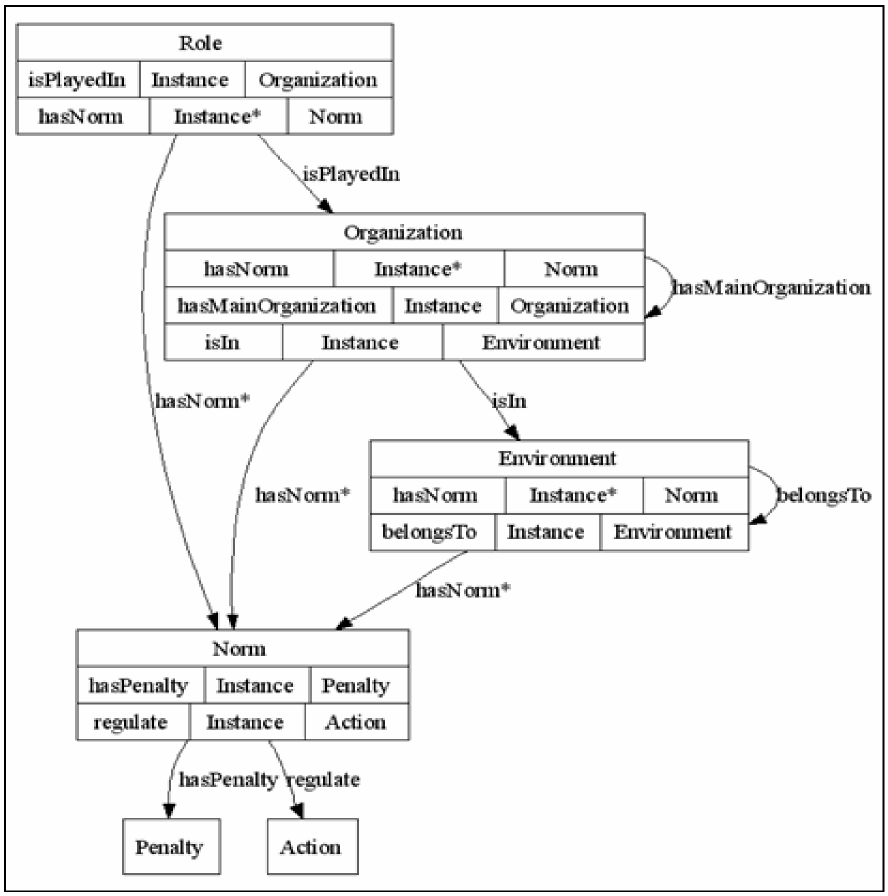

Figure 3. The DynaCROM contextual normative meta-ontology

The DynaCROM contextual normative meta-ontology (or, simply, the DynaCROM meta-ontology) is constituted by six related concepts. The *Action* concept encompasses all instances of the regulated actions. The *Penalty* concept encompasses all instances of the fines to be given in case norms are not fulfilled. The *Norm* concept encompasses all norm instances from all regulatory contexts, and each norm instance represents a permission, obligation or prohibition for its associated action and penalty. The *Environment* concept encompasses all instances of the regulated environments, and each environment encompasses its associated norms and its owner environment (the environment it belongs to). The *Organization* concept encompasses all instances of the regulated organizations, and each organization encompasses its associated norms, main organization (the organization which it is associated to) and environment. The *Role* concept encompasses all instances of the regulated roles, and each role encompasses its associated norms and organization.

While regulating open MAS from particular domains, the DynaCROM meta-ontology should be instantiated with modeled laws and it should be probably extended with both domain concepts and interaction laws. Interaction laws should be implemented by following a representation pattern, from the *Semantic Web Best Practices* document [Noy and Rector 2006]. This pattern defines that the relation object itself must be represented by a created concept that links the other concepts from the relation (i.e., reification of the relationship). Thus, in DynaCROM ontologies, an interaction law is represented by a new Norm sub-concept that links two Role concepts.

3. Regulating an Open MAS with DynaCROM

The domain of multinational corporations is chosen to explain how to implement contextual regulations in an open MAS with DynaCROM. A multinational corporation (organization) is an enterprise that manages production branches located in at least two countries. These branches can be in different regions across multiple continents. Corporate regulations include control all possible relationships among the many players involved.

Hpie is our created main organization and it has Hpie Cuba and Hpie Brazil as its branches. Hpie corporations have the following roles: manufacturer, supplier, distributor and customer. Hpie is in USA, which in turn is in North America; Hpie Cuba is in Cuba, which in turn is in Central America; and Hpie Brazil is in Brazil, which in turn is in South America.

3.1. Modeling Contextual Laws

Corporation laws are usually private because they are strategic for the corporation businesses. So, we created environment, organization, role and interaction laws based on public laws collected from several corporate Web sites. These laws were classified according to the DynaCROM top-down modeling of contextual laws.

1. **Examples of Environment Laws:**

   1.1. In North America, a finished good from every organization is obliged to have its price increased by a percentage (dependent of the seller location) as taxes, for immediate delivery or if the deliver address is in North America.

   1.2. In USA states, a finished good from every organization is obliged to have its price increased by 8% as taxes, for immediate delivery or if the deliver address is in an USA state.

   1.3. In South America, if the deliver address is outside there, every shipped order is obliged to have its price increased by 15% as taxes.

   1.4. In USA, all negotiations are obliged to be paid in American dollars (USD), its national currency. Negotiations outside USA are obliged to have their values converted from USD to the national currency of the country where the seller is.

1.5. In Cuba, all negotiations are obliged to be paid in Cuban pesos (CUP), its national currency. Negotiations outside Cuba are obliged to have their values converted from CUP to the national currency of the country where the seller is.

1.6. In Brazil, all negotiations are obliged to be paid in Reais (R$), its national currency. Negotiations outside Brazil are obliged to have their values converted from R$ to the national currency of the country where the seller is.

2. **Examples of Organization Laws:**

2.1. In Hpie, all paid orders are obliged to have detailed receipts.

2.2. In Hpie Cuba, every product is obliged to have one year of warranty.

2.3. In Hpie Brazil, every placed order is obliged to have a down payment of 10%.

3. **Examples of Role Laws:**

3.1. Hpie Cuba manufacturers are obliged to provide refunds or replacements for every defective product when substantial defects cannot be fixed in four attempts.

3.2. Hpie Cuba manufacturers are obliged to provide, with one month, refunds or replacements for every defective product, when substantial defects cannot be fixed in four attempts.

3.3. Hpie Brazil suppliers are obliged to ship orders in their due dates.

3.4. Hpie Brazil suppliers are permitted to give 5% discount for orders paid in cash.

4. **Example of an Interaction Law:**

4.1. Hpie Brazil suppliers are permitted to ship incomplete orders to manufacturers.

3.2. Instantiating a DynaCROM Ontology for a multinational domain

Figure 4. illustrates a DynaCROM ontology instance, representing the entities created for our case study. North America, Central America, South America, USA, Cuba and Brazil are instances of the *Environment* concept. Hpie, Hpie Cuba and Hpie Brazil are instances of the *Organization* concept. Hpie manufacturer, supplier, distributor and customer roles are instances of *Role* sub-concepts, which extend the DynaCROM meta-ontology and are instantiated for each organization. Two examples of role instances are "*AHpieBrazilSupplier*" from Hpie Brazil and "*AHpieCubaManufacturer*" from Hpie Cuba. Both roles are linked by the "*PermissionToShipIncompleteOrders*" norm and the triple represents the interaction law 4.1 (illustrated by the dashed line in Figure 4).

Other contextual laws are also created as instances of the Norm concept and are attached to their respective instances. Some examples are: the environment law 1.1. attached to North America; the environment law 1.4. attached to USA; the organization law 2.1. attached to Hpie; the organization law 2.3. attached to Hpie Brazil; and the role law 3.3. attached to the Hpie Brazil supplier. All these laws are also illustrated in Figure 4. All actions regulated by the presented laws are instances of the *Action* concept and are associated to their respective penalties.

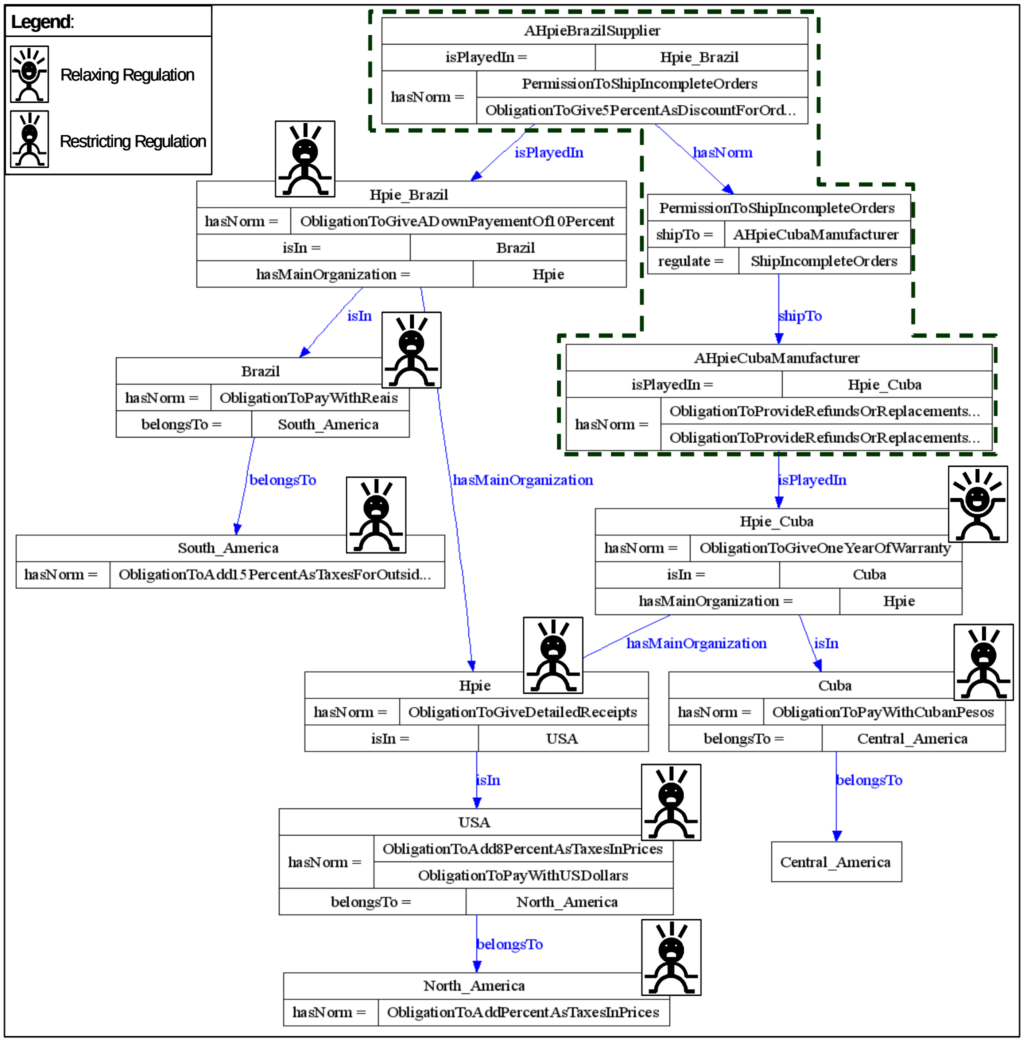

Figure 4. Contextual domain laws represented in a DynaCROM ontology instance

3.3. Dynamically Restricting and Relaxing Contextual Laws

The main asset of organizing laws following a top-down modeling is to permit flexibility while enforcing different customized compositions of contextual laws. By doing so, the system regulation can be dynamically relaxed or restricted. Returning to our example, in different situations the current system regulation was relaxed and restricted by eight compositions of contextual laws, as illustrated with the particular icons from Figure 4.

An example of law relaxation is Hpie Cuba manufacturers being also regulated through the Hpie Cuba organization law 2.2 (stating that warranty period is limited to one year). Examples of law restriction are: Hpie Brazil suppliers being also regulated through the Hpie Brazil organization law 2.3 (stating that every placed order must have a down payment of 10%); Hpie Brazil being also regulated through the Brazil environment law 1.6. (stating that all negotiations in Brazil must be paid with Reais); Brazil being also regulated through the South America environment law 1.3. (stating that every shipped order to deliver addresses outside South America must have its price increased by 15% as taxes); and, finally, Hpie Brazil being also regulated through the Hpie organization law 2.1 (stating that all paid orders must have detailed receipts).

Many other compositions of contextual laws can be done, according to the DynaCROM ontology structure (i.e., according to the relations between its concepts). However, instead of implement the desired compositions of contextual laws inside agent or system codes and have to change these codes every time different compositions of contextual laws are required, DynaCROM offers a more flexible solution. Based on a DynaCROM ontology instance (which expresses laws) and on sets of rules (which defines compositions of contextual laws), DynaCROM uses a rule-based inference engine to automatically deduce the current composition of contextual laws an agent is bound to, while playing a regulated action (see this process in Figure 1).

DynaCROM rules are ontology-based rules, i.e. they are created according to the ontology structure and they are limited according to the number of related concepts each concept is linked to. For instance, in Table 1, when **rule 1** is activated, a given organization has its norms composed with its main organization norms (e.g., Hpie Brazil norms are composed with Hpie norms); when **rule 2** is activated, a given organization has its norms composed with its environment norms (e.g., Hpie Brazil norms are composed with Brazil norms); when **rules 1 and 2** are activated, a given organization has its norms composed with both their main organization and environment norms (e.g., Hpie Brazil norms are composed with both Hpie norms and Brazil norms); when **rule 3** is activated, a given organization has its norms composed with the norms from its environments (e.g., Hpie Brazil norms are composed with South America norms).

Rules can compose norms from directly or indirectly related concepts, and from the same or different types. For instance, Table 1 illustrates compositions of norms from directly related concepts from the same type (e.g., rule 1 composes organization and main organization norms); from directly related concepts from different types (e.g., rule 2 composes organization and environments norms); and from indirectly related concepts from different types (e.g., rule 3 composes organization norms with the norms of the owner environments of their environments).

Table 1. Rules for compositions of contextual laws

```
Rule1- [ruleForOrgWithMOrgNorms:
        hasNorm(?Org,?MOrgNorms)
          <- hasNorm(?MOrg,?MOrgNorms),
             hasMainOrganization(?Org,?MOrg)]

Rule2- [ruleForOrgWithEnvNorms
        hasNorm(?Org,?OrgEnvNorms)
          <- hasNorm(?OrgEnv,?OrgEnvNorms),
             isIn(?Org,?OrgEnv)

Rule3- [ruleForOrgWithOEnvOfOrgEnvNorms
        hasNorm(?Org,?OEnvOfOrgEnvNorms)
          <- hasNorm(?OEnvOfOrgEnv,?OEnvOfOrgEnvNorms),
             belongsTo(?OrgEnv,?OEnvOfOrgEnv)
             isIn(?Org,?OrgEnv)]

Rule4- [ruleForOrgWithMOrgEnvNorms:
        hasNorm(?Org,?MOrgEnvNorms)
          <- hasNorm(?MOrgEnv,?MOrgEnvNorms),
             isIn(?MOrg,?MOrgEnv)
             hasMainOrganization(?Org,?MOrg)]
```

```
Rule5- [ruleForOrgWithOEnvOfMOrgEnvNorms:
        hasNorm(?Org,?OEnvOfMOrgEnvNorms)
        <- hasNorm(?OEnvOfMOrgEnv,?OEnvOfMOrgEnvNorms),
           belongsTo(?MOrgEnv,?OEnvOfMOrgEnv)
           isIn(?MOrg,?MOrgEnv)
           hasMainOrganization(?Org,?MOrg)]
```

3.4. The DynaCROM Implementation

The DynaCROM implementation is divided into three parts: the contextual normative meta-ontology created using the Protégé ontology editor [Stanford University 2006], the rules written according to the Jena API rule syntax [Hewlett-Packard Development Company 2006] and the implementation of a JADE behavior [Tilab Company 2006], encapsulating the DynaCROM approach.

Table 2 shows part of the code of the DynaCROM behavior. The result of the "getOntModel()" method (see line 9) is a DynaCROM ontology instance, which explicitly represents the user defined regulatory contexts and laws. Activations and deactivations of rules, used to define new compositions of contextual laws, must have to be written in the "rulesToComposeNorms.rules" file (see line 4). The "reasoner" variable (see line 5) represents the rule-based inference engine which, based on the ontology and active rules, automatically deduces the customized compositions of contextual laws and keeps these laws in the "inferredModel" variable (see line 8).

Table 2. Applying compositions of contextual laws into an ontology instance

```
(1) Model m = ModelFactory.createDefaultModel();
(2) Resource configuration =  m.createResource();
(3) configuration.addProperty(ReasonerVocabulary.PROPruleSet,
(4)   ontologyDir.concat("rulesToComposeNorms.rules"));

(5) Reasoner reasoner =
(6) GenericRuleReasonerFactory.theInstance().create(configuration);

(8) InfModel inferredModel =
(9) ModelFactory.createInfModel(reasoner, this.getOntModel());
```

Developers of regulations in open MAS, aiming at the use of our DynaCROM approach, should complete the following four steps: (1) classify and organize their laws according to the DynaCROM top-down modeling; (2) extend the DynaCROM meta-ontology with particular domain concepts and explicitly represent all modeled laws into this extended ontology; (3) write rules, according to their domain concepts, and activate those (eliminating comments) or deactivate those (adding comments), for the automatic compositions of contextual laws from their particular domain; and, finally, (4) enhance their agents and systems with the DynaCROM behavior (totally free) for the enforcement of composed contextual laws or implement a similar behavior.

The implementation of our case study followed the above steps, i.e. we classified and organized our laws according to the DynaCROM top-down modeling and explicitly represented them in an extended instance of the DynaCROM meta-ontology. Our agents were implemented in JADE [Tilab Company 2006] and enhanced with the two behaviors: a migratory behavior, which made them move randomly from one location (environment or organization) to another; and the DynaCROM behavior, which is responsible for enforcing the user-defined customized compositions of contextual laws. The dynamic of DynaCROM for law evolution was perceived while creating, deleting and updating laws into the created ontology instance and while activating and deactivating new sets of rules for different customized compositions of contextual laws. Figure 5. illustrates part of the world of our implemented case study, where environments and organizations are represented by JADE containers, and the "****MobileAgent*", found in Hpie Brazil, is an example of an agent which has the migratory and DynaCROM behaviors.

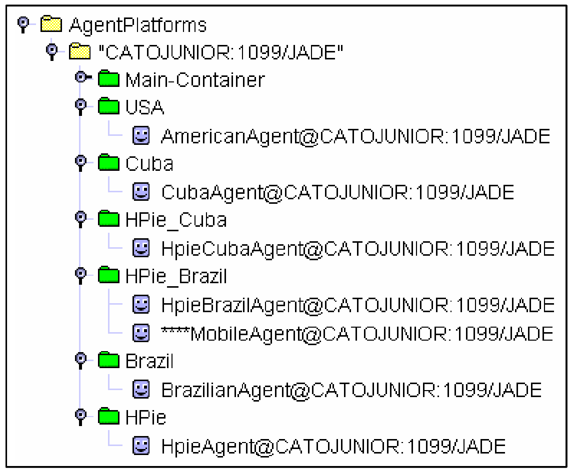

Figure 5. An open MAS regulated with DynaCROM

4. Related Work

[García-Camino *et al.* 2005] proposes a distributed architecture to endow MAS with a social layer, in which normative positions are explicitly represented and managed via rules. Every external agent from the architecture has a *dedicated governor agent* connected to it, enforcing the laws of executed events. DynaCROM also uses rules to manage normative agent positions, but executed actions, instead of executed events, are the focus of the regulation. Besides, DynaCROM provides a more precise mechanism for regulation, while enforcing customized compositions of contextual laws. Furthermore, enforcement can be done with few governor agents responsible for monitoring only the system regulated actions, instead of having many agents monitoring all events executed in the regulated system.

[Vázquez-Salceda *et al.* 2005] proposes the OMNI (Organizational Model for Normative Institutions) framework for modeling agent organizations. Comparing DynaCROM with OMNI, both define a meta-ontology with a taxonomy for norm regulations in open MAS. 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 DynaCROM it can be carried out by trusted agents or by specific regulatory mechanisms from the regulated systems. A second and more important difference 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 regulatory contexts. DynaCROM is based on laws for the environment, organization, role and interaction contexts, to simplify the enforcement and evolution processes. For instance, the social structure of an organization in OMNI describes, at 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. DynaCROM uses the organization regulatory context to specify organization norms that hold for all roles from an organization and it uses the role regulatory context to specify role norms, both regulatory contexts from different levels of abstractions.

5. Conclusion

In this paper, we presented our ongoing work for dynamic contextual regulations in open MAS – called DynaCROM – and how to use DynaCROM for particular domains. DynaCROM is a straightforward method to smoothly apply and manage regulations that can be dynamically relaxed or restricted by compositions of contextual laws. However, the current implementation of DynaCROM does not prevent norm-aware agents from executing actions that violate norms; it only penalizes infringing agents for doing so. Thus, the agent autonomy is preserved leaving for them to decide whenever obeys laws. Moreover, DynaCROM still doesn't make any assumption about the problem of how resolve raised conflicts when laws state the same subject in an opposite way.

For future work, we are currently studying four main research lines: (1) contexts and context-aware systems, (2) use of action ontologies, (3) simulations of regulated open MAS and (4) use of third-party libraries of agent behaviors. The idea is to explore independently each of these research lines and to enhance DynaCROM, if good results appear. Dealing with contexts (1), we expect to better understand DynaCROM precision in regulations. Thus, we are working on an example where a DynaCROM physical platform (e.g. a notebook) changes places, instead of its agents. Refining the granularity level of contexts, laws can be applied, for example, to particular rooms [Viterbo et al. 2006]. Dealing with the use of action ontologies (2), we expect to identify the minimal *a priori* knowledge that agents must have to deal with DynaCROM and, consequently, get its benefits. We know that the action and *(DynaCROM)* normative ontologies are deeply related. For this phase, we are only concerned with which attributes and related concepts an action should have. Next, we will try to represent (maybe with planner tools) the causes and effects of actions. Dealing with simulations (3), we expect, with a graphic engine, to easily change some particular environment variables and, after this, analyze agent reactions ahead several (unpredictable) situations. Finally, dealing with third-party libraries of agent behaviors (4), we expect better options for agents adapt themselves ahead different unexpected situations.

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

Artikis, A.; Pitt, J. and Sergot, M. (2002) "Animated specifications of computational societies". In Proc. of AAMAS-2002, Part III, pages 1053-1061, Italy.

Bouquet, P.; Giunchiglia, F.; van Harmelen, F.; Serafini, L.; and Stuckenschmidt, H. (2003) "C-OWL: Contextualizing Ontologies". In Proc. of ISWC-2003, LNCS vol. 2870, p. 164-179, Springer Verlag.

Castelfranchi, C.; Dignum, F.; Jonker, C.M. and Treur, J. (1999) "Deliberative Normative Agents: Principles and Architecture". In Proc. of ATAL-1999, USA.

Dey, A. (2001) “Understanding and Using Context. Personal and Ubiquitous Computing”. Springer-Verlag, London, UK; Vol. 5, No.1, p. 4-7.

Felicíssimo, C.; Lucena, C.; Briot, J.-P.; Choren, R. (2006) “An Approach for Contextual Regulations in Open MAS”. In Proc. of AOIS@AAMAS-2006. Japan.

Ferber, J.; Gutknecht, O.; Michael, F. (2003) “From Agents to Organizations: an Organization View of Multi-Agent Systems”. In Proc. of the AOSE, 2003. Australia.

García-Camino, A.; Rodrígurez-Aguilar, J.A.; Sierra, C. and Vasconcelos, W. (2005) “A Distributed Architecture for Norm-Aware Agent Societies”. In DALT 2005.

Grizard, A.; Vercouter, L.; Stratulat, T.; and Muller, G. (2006) “A peer-to-peer normative system to achieve social order”. In Proc. of COIN@AAMAS-2006. Japan.

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

Hewitt, C. (1991) “Open Information Systems Semantics for Distributed Artificial Intelligence” AI, 47(1-3), p. 79-106. ISSN: 0004-3702.

Hewlett-Packard Development Company (2006). Jena. In <http://jena.sourceforge.net/>.

Jennings, N.; Sycara, K. and Wooldridge, M. (1998) “A Roadmap of Agent Research and Development”. Journal of Agents and Multi-Agent Systems, 1:p.7-38.

Jennings, N. R. (2000) “On Agent-Based Software Engineering”. AI, 117(2) p.277-296.

Khedr, M; Karmouch, A. (1995) “ACAI: Agent-Based Context-aware Infrastructure for Spontaneous Applications”. Journal of Network & Computer Applications, Vol. 28, Issue 1, p. 19-44.

Noy, N; Rector, A. (2006.). Eds. Semantic Web Best Practices and Deployment Working Group: Defining N-ary Relations on the Semantic Web: Use with Individuals. In <http://www.w3.org/TR/swbp-n-aryRelations/>.

OMG Agent Working Group. (1999) “Agent Technology Green Paper”. In: <http://www.objs.com/agent/agents_Green_Paper_v100.doc>.

Stanford University (2006) Protégé. In <http://protege.stanford.edu/>.

Thomas, G. and Williams, A. B. (2005) “Roles in the Context of Multiagent Task Relationships”. AAAI Fall Symposium “Roles, an Interdisciplinary Perspective: Ontologies, Programming Languages, and Multiagent Systems”, FS-05-08. ISBN 978-1-57735-254-9. USA.

Tilab Company. (2006) JADE. In: <http://jade.tilab.com/>.

Vázquez-Salceda, J.; Dignum, V. and Dignum, F. (2005) “Organizing Multiagent Systems”. Journal of Autonomous Agents and Multi-Agent Systems, 11, p. 307-360.

Viterbo, J.; Felicíssimo, C.; Briot, J.-P.; Endler, M.; Lucena, C. (2006) “Applying Regulation to Ubiquitous Computing Environments”. In: (SEAS 2006).

Weyns, D.; Parunak, H.V.D.; Michel, F.; Holvoet, T. and Ferber, J. (2005) “Environments for Multiagent Systems State-of-the-Art and Research Challenges”. In Proc. of the E4MAS, 2005, p.1-47. ISBN 3-540-24575-8.