PUC

ISSN 0103-9741

Monografias em Ciência da Computação

n° 32/05

Sweetening Regulated Open Multi-Agent Systems with Support for Agents to Reason about Laws

Carolina Howard Felicíssimo

José Alberto Rodrigues Pereira Sardinha

Carlos José Pereira de Lucena

Departamento de Informática

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22453-900

RIO DE JANEIRO - BRASIL

Monografias em Ciência da Computação, No. 32/05 ISSN: 0103-9741
Editor: Prof. Carlos José Pereira de Lucena October, 2005

Sweetening Regulated Open Multi-Agent Systems with Support for Agents to Reason about Laws *

Carolina Howard Felicíssimo, José Alberto Pereira Rodrigues Sardinha,

Carlos José de Pereira de Lucena

(cfelicissimo, sardinha, lucena)@inf.puc-rio.br

Abstract. Inspired by the Tim Berners-Lee's vision of a Semantic Web, this work aims to present an approach to regulate open Multi-Agent Systems (MAS) based on ontologies plus layers of logic and rules for data inference. In our approach, a top-down modeling of laws is designed for regulating an open MAS. We propose to regulate agents' actions based on four levels of abstractions: Environment laws, organization laws, role laws and interaction laws. These levels of regulations are represented by a domain independent normative ontology, which has the six related main concepts: Environment, Organization, Role, Action, Norm and Penalty. The layer of logic from our solution is composed of a combination of Description Logic, Deontic Logic and some ideas from Defeasible Logic. The rule layer is composed of user defined rules and a rule-based inference engine. Some advantages of this work are: automatic composition of laws by using rules and inference, facility to regulate agents' actions considering all the designed law levels, consistency check for pre-defined laws, and support for black-boxes agents to reason about laws in open MAS.

Keywords: Ontologies and Agent Systems, Logics for Agent Systems.

Resumo. Inspirado pela visão de Tim Berners-Lee da Web Semântica, este trabalho vias apresentar uma aproximação para regular Sistemas Multi-Agentes (SMA) abertos baseado em ontologias mais camadas de lógica e regras para inferência de dados. Em nossa aproximação, uma modelagem *top-down* de leis é projetada para regulamentação de SMA abertos. Nós propomos regular ações de agentes baseados em quatro níveis de abstrações: *Leis de Ambiente*, *Leis de Organização*, *Leis de Papel* e *Leis de Interação*. Estes níveis de regulamentação são representados por uma ontologia normativa independente de domínio, a qual possui os seis conceitos principais relacionados: *Environment, Organization, Role, Action, Norm* e *Penalty*. A camada lógica da nossa solução é composta pela combinação de *Description Logic, Deontic Logic* e algumas idéias de *Defeasible Logic*. A camada de regras é composta por regras definidas por usuários e uma máquina de inferência baseada em regras. Algumas vantagens desse trabalho são: composição automática de regras pela utilização de regras e inferência, facilidade de regular ações de agentes considerando todos os níveis de leis projetados, checagem de consistência para leis pré-definidas, e suporte para agentes caixas-pretas raciocinarem sobre leis em SMA ambientes.

Palavras-chave: Ontologias e Sistemas Agentes, Lógica para Sistemas Agentes

---

* Trabalho patrocinado pelo Ministério de Ciência e Tecnologia da Presidência da República Federativa do Brasil (e agência de fomento e o número do processo, se aplicável). (Em Inglês: This work has been sponsored by the Ministério de Ciência e Tecnologia da Presidência da República Federativa do Brasil)

In charge for publications:

Rosane Teles Lins Castilho
Assessoria de Biblioteca, Documentação e Informação
PUC-Rio Departamento de Informática
Rua Marquês de São Vicente, 225 - Gávea
22453-900 Rio de Janeiro RJ Brasil
Tel. +55 21 3114-1516 Fax: +55 21 3114-1530
E-mail: bib-di@inf.puc-rio.br
Web site: http://bib-di.inf.puc-rio.br/techreports/

Table of Contents

1 INTRODUCTION

Agents are autonomous flexible entities acting in order to meet their design objectives – they are, in short terms, goal-oriented entities [24]. Agents play in Multi-Agent Systems (MAS), which are typically open (agents can get in and out) and have no single centralized designer [23]. Agents are self-governed autonomous entities that usually interact based only on their own beliefs and capabilities [1], and with some intended individual or a collective purpose. Openness [21] has led to software systems that have no centralized control and that are composed of autonomous entities [2], as agents. Key characteristics of open MAS are: agent heterogeneity, conflicting individual goals and limited trust [4]. The design of open systems is a new paradigm in the software development field.

In the development of an open MAS, regulation of agents' actions is a very important aspect in the system's design. To regulate agents' action in an efficient way, some issues have to be addressed: (i) how to express laws in a meaningful and precise way for agents to understand those, (ii) how to provide a semantic support for agents to decide what to do, and (iii) how an open MAS can be regulated by laws.

In (i), laws have to be addressed even when they are expressed in different notations. For instance, an environment of an open MAS can be composed of three countries: Brazil, USA and Germany. Agents in this environment can move around from one country to the others as many times they want. However, different countries have same laws, which are normally expressed in distinct ways. Moreover, agents have to obey and understand the laws of each country whenever they migrate from one place to another one. In the urban traffic domain, traffic signs express the laws. Figure.1 illustrates different law representations by country: in the first line, we depict the "Pedestrians prohibited" sign in Germany (left side) and in Brazil (right side); in the second line, we depict the German sign of "Recommended speed limit" (left side) and the "Speed limit" sign in Brazil (right side); and in the third line, we depict the "Road closed" signs in Germany (left side) and in USA (right side). Human drivers have to read from books these urban traffic laws in order to learn the meanings of the traffic signs and how to act according to them. However, we cannot assume that heterogeneous agents are able to automatically understand laws in different languages and in different notations. Laws for agents have to be written in a common language in order to provide precise information for machine understanding.

Figure.1. Different traffic signs by country

In (ii), laws have to be addressed to agents decide which action to perform, such as the selection of actions that are permitted, obligated and prohibited. Autonomous normative agents are entities that are able to take into account the existence of social norms in their decisions (either to follow or violate a norm) and are able to react to the violations of norms by other agents [11]. Normative agents have a semantic support to decide, for example, what to do when unpredictable situations happen. For instance, Figure.2 illustrates such situation: an agent is driving in a road and another agent is in his back side. Suddenly, a cow appears in front of his car. What should the agent do? There is a clear tradeoff in the selection of his action: kill the cow or crash the car? This decision has to be based on the consequences of the selected action, i.e. the action that has the smallest penalty.

Figure.2. An example of an unpredictable situation

In (iii), this third issue has to be addressed after the issues (i) and (ii) have been addressed. There is a clear dependency of this third issue with the first two others, e.g. to regulate an open MAS with laws it is necessary to express laws in a meaningful and precise way for agents to understand those, and to provide a semantic support for the selection of the actions by agents.

Seeking the identification of a taxonomy for regulations in an open MAS, issues (i), (ii) and (iii) have been addressed in previous works [16] and [17], and continue being addressed in this work, as an evolution of the two others. However, this work also proposes some guidelines for regulating an open MAS based on ontologies plus layers of logic and rules for agents to reason about laws. Our solution is inspired by the Tim Berners-Lee's vision of a Semantic Web [8], where the "Ontology support" layer is the base of the "Logic" and "KR rules" layers, according to his "Semantic Web Layer Cake design" [9], illustrated in Figure.3.

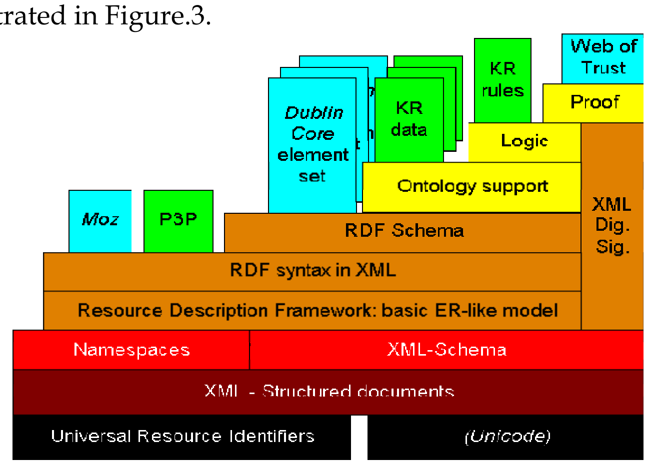

Figure.3. The Semantic Web Layer Cake design, from [9]

The aim of this paper is to describe how the regulation of an open MAS can be sweetened when the reasoning about laws is required for agents. The paper is organized as follow: the next section describes our guidelines for regulations in open MAS with a formal support; section 3 presents a case study from the supply chain domain that was implemented to validate our solution; section 4 briefly discussed some related work; and, finally, the conclusions and future works are presented in Section 5.

2 GUIDELINES FOR REGULATIONS IN OPEN MAS WITH A FORMAL SUPPORT

A MAS is composed, mainly, of environments, organizations, agents and agents' interactions [24]. Environments [34] are computational infrastructures that provide the conditions for agents to inhabit it. An environment can be composed of many organizations, i.e. partitions and groups of entities such as departments, communities and societies [18]. An organization from a MAS is composed of a group of agents playing roles in it. Roles characterize the positions of agents in organizations. Organizations define roles and can also define sub-organizations. At the same time, an agent can belong to many organizations from only one environment [30]. However, agents with the mobility characteristic can move from one environment to another or can register or leave organizations, obeying or not their defined laws.

To decrease the difficulty of implementing regulations in open MAS, a top-down model of laws is defined. In this approach, an evolution of [16] and [17], the four levels of abstractions are proposed: (i) environment laws, (ii) organization laws, (iii) role laws and (iv) interaction laws. Environment laws are those that are applied to all agents from the regulated environment, independently of its organizations, roles and interactions. Organization laws are those that are applied to all agents from the regulated organization, independently of its roles and interactions. Role laws are those that are applied to all agents playing the regulated role, independently of its interactions. Interaction laws are those that are applied to all agents involved in the regulated interaction.

Norms can regulate environments, organizations, agents' roles and interactions, and, consequently, can control the actions performed in an open MAS defining which are permitted, obligated and prohibited. A permitted norm defines that an act is allowed to be performed; an obligatory norm defines that an act must be performed; a prohibited norm defines that an act must not be performed. The three types of norms described represent the three fundamental deontic statuses of an act [3] from Deontic Logic [34]. Deontic Logic enables to address the issue of explicitly and formally defining norms and dealing with their possible violation.

To provide norms regulation according to the Deontic Logic and consciousness of agents in open MAS, a semantic support is desired. This type of support can be given by ontologies, making the represented information of a domain easier for machines to automatically process their meanings [25].

2.1 A Domain Independent Normative Ontology to Assist Regulations in Open MAS

Ontologies are conceptual models that embody shared conceptualizations of a given domain [20]. Ontology languages are designed to be used by applications (machines) that need to process the content of information instead of just presenting information to humans [31].

Normative ontologies are those that have the norm concept as a central asset. This kind of ontology provides information for norm-autonomous agents, committed to their roles, guide their behaviors on, for example, goals and plans. A norm-autonomous agent capable of obtaining the semantic support provided by a normative ontology (its internal structure accesses ontologies) is an agent with intelligence for norm violation by action selection mechanisms. Action selection mechanisms [10], [33] permit effectiveness for agents to achieve their goals.

In this work, we propose a domain independent normative ontology, designed in order to assist regulations in open MAS. The ontology restricts, with norms and their associated penalties, agents' actions by the four levels of regulations: environment laws, organization laws, role laws and interaction laws. The three first levels are represented in the ontology by the six related main concepts (illustrate in Figure.4), all in the same hierarchical level: Environment, Organization, Role, Norm, Penalty and Action. The representation of interaction laws is dependent of domains and, because of this, it is not presented in the domain independent ontology.

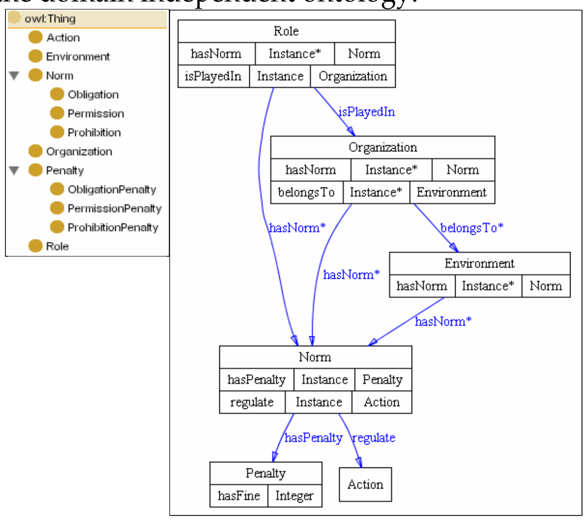

Figure.4. A domain independent normative ontology

The six related domain independent concepts from our approach have specific data associated with them. The *Environment* concept holds its norms. The *Organization* concept holds its norms and all environments where the organization belongs to. The *Role* concept holds its norms and the organization where it can be played in. The *Norm* concept holds its associated penalties, to inhibit norm violation, and its regulated actions. The *Penalty* concept holds a fine to be given if its associated norm is violated. The *Action* concept holds the actions that must be regulated because those bring effects to environments, organizations or other agents. The Norm and Penalty concepts are specialized in sub-concepts according to the permitted, obligated and prohibited statuses of an act from Deontic Logic.

As written before, the level of interaction laws is dependent of domains and, because of this, its representation is not presented in the domain independent ontology. However, interaction laws must be implemented during the ontology extension and instantiation processes by following the representation pattern from the *Semantic Web Best Practices* document [26]. This pattern defines that the relation object itself is represented by a created concept and it has to be linked to the others concepts from the relation. For instance, in interaction laws' representations, the created object is a sub-concept from the Norm concept and it has to be linked to Role concepts. The representation pattern is suggested when the relation between concepts are relevant and strong, has associated information with it and more than one participant needs to be addressed explicitly. For instance, Figure.5 illustrates how the same relation between A and B, A and C, B and C, and vice-versa should be designed according to the pattern. The box with the "P" letter represents the created relation object.

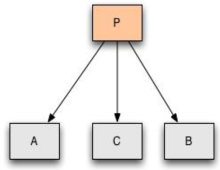

Figure.5. Defining N-ary relations, from [26]

With our top-down modeling of laws at different levels for regulation and the ontology support for such representations, interactions considering all related defined law levels can be regulated easily. An interaction is among instances of roles; each role has its norms and its organization; each organization has its norms and its environments; each environment has its norms. In all the regulated levels, norms can be accessed by a backward chaining. Agents have to act compliant with all their defined laws, otherwise, penalties from violated norms are given to them. A scenario where interactions are regulated considering all the defined law levels is illustrated in Figure.6, which has some interactions represented in it by arrows.

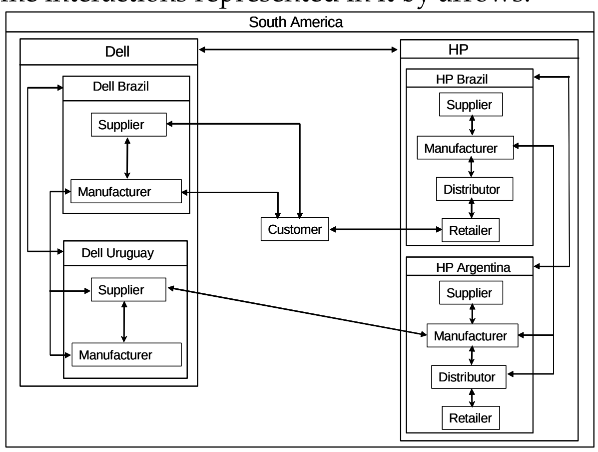

Figure.6. Examples of regulated interactions

In Figure.6, "South America" is an environment; "Dell" and "HP" are main organizations; "Dell Brazil" and "Dell Uruguay" are sub-organizations from "Dell"; "HP Brazil" and "HP Argentina" are sub-organizations from "HP"; "Supplier" and "Manufacturer" are Dell roles; and, finally, "Supplier", "Manufacturer", "Distributor" and "Retailer" are HP roles. Agents from the Dell and Hp organizations while interacting can be regulated by the norms from Dell (for agents from Dell), HP (for agents from HP) and South America. Suppliers and manufacturers from Dell Brazil while interacting can be regulated by the norms from Dell Brazil, Dell and South America. Manufacturers from Dell Brazil while interacting with manufacturers from Dell Uruguay can be regulated by the norms from Dell Brazil, Dell Uruguay, Dell and South America. Suppliers from Dell Uruguay while interacting with manufacturers from HP Argentina can be regulated by the norms from Dell Uruguay, HP Argentina, Dell, HP and South America. All the concepts from the example have information to permit a backward chaining to access their associated norms.

2.2 A Logic Support to Assist Regulations in Open MAS

A logic layer is designed in our approach by a combination of Description Logic [6], Deontic Logic [34] and some ideas from Defeasible Logic [27] in order to assist regulations in open MAS for agents to reason about laws. Description Logic is added to the ontology layer by writing OWL-DL [7] normative ontologies. OWL-DL is an ontology type that supports reasoners for Description Logic. Deontic Logic is used in the logic layer to define permitted, obligated and prohibited actions as sub-concepts of the Norm concept from the normative ontology of our approach. The ideas from Defeasible Logic are concentrated in how its theory can resolve the existing conflicts when the laws from the organization and environment levels have to be combined, i.e. sometimes it is not desire that all norms from all organization's environments be added to the organization laws.

Defeasible Logic theory is composed of five elements: Facts (indisputable statements), Strict Rules (whenever the premises are indisputable, e.g. facts, then so is the conclusion), Defeasible Rules (rules that can be defeated by contrary evidence), Defeaters (rules that cannot be used to draw any conclusions, their only use is to prevent some conclusions. They are used to defeat some defeasible rules by producing evidence to the contrary) and a Superiority Relation among Rules (used to define priorities among rules, i.e., where one rule may override the conclusion of another rule) [27].

Defeasible Logic can be successfully used to resolve conflicts by using the superiority relation among rules and to just fire rules of the ideal combination of organization laws and environment laws. For instance, Figure.7 illustrates a conflict situation when the Dell organization inherits all its environment laws from South America, Brazil and Uruguay. In South America is prohibited to ship incomplete orders (a defeasible rule), but in Brazil is permitted to do that (a defeater). Using the Defeasible Logic ideas to resolve the conflict, a rule can be written specifying that Brazil and Uruguay are environments more specific than South America and, because of this, their rules have greater priority than the South America ones. In this way, the conflict is resolved by letting incomplete order to be shipped in Brazil.

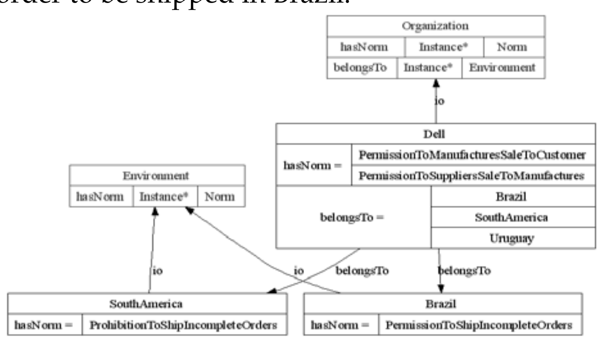

Figure.7. An example for use the Defeasible Logic ideas

Defeasible Logic can also be used when agents from different organizations of the same environment are interacting and it is desire that only the laws from the same environment are inherited, without the laws from the others organizations' environments. For example, during interactions between agents from the Dell and HP organizations in Uruguay, it is not necessary/desirable that norms from others environments, like the Brazil one, are inherited. In this case, the ideal combination of organization laws and environment laws are the Dell, HP, Uruguay and South America norms without the norms from Brazil. In this case, Defeasible Logic can be used to specify that only the laws from the same environment of the organizations interacting will be inherited, e.g. only norms from Uruguay and South America have to be inherited.

2.3 A Rule Support to Assist Regulations in Open MAS

Rules can be defined for data inference and to assist regulations in open MAS for agents to reason about laws. The rule layer has rules defined based on the normative ontology taxonomy, i.e. rules can be specified considering the defined concepts and relations from the normative ontology. From specified rules and by using a rule based inference engine [29], data can be inferred. For instance, Table 1 has three rules specified (r1, r2 and r3) representing that role's norms can also have the norms from the organization where the role is played in (r2) and the norms from the environments where the organization belongs to (r3). The concepts "Role", "Organization" and "Environment", and the relations "hasNorm", "isPlayedIn" and "belongsTo" are specified in the normative ontology. With the rules specified and the ontology filled with data (instances), if a question about "what are the norms of a role?" is proposed to the inference engine, the answer will follow the steps: a role has norms; for each organization where the role is played in, the role also has its organizations' norms (r2); for each organization where the role is played in and for each environment where the organizations belongs to, the role also has its environments' norms (r3).

Table 1. Examples of rules for normative ontologies

```
@prefix tacUri: <http://www.owl-ontologies.com/tac.owl#>.
@include <RDFS>.

-> table(rdfs:subClassOf).

[r1: (?A rdfs:subClassOf ?C)
        <- (?A rdfs:subClassOf ?B) (?B rdfs:subClassOf ?C)]

[r2: (?Role tacUri:hasNorm ?RoleNorm)
     (?Role tacUri:isPlayedIn ?Organization)
     (?Organization tacUri:hasNorm ?OrganizationNorm)
        -> (?Role tacUri:hasNorm ?OrganizationNorm)]

[r3: (?Role tacUri:hasNorm ?RoleNorm)
     (?Role tacUri:isPlayedIn ?Organization)
     (?Organization tacUri:hasNorm ?OrganizationNorm)
     (?Organization tacUri:belongsTo ?Environment)
     (?Environment tacUri:hasNorm ?EnvironmentNorm)
        -> (?Role tacUri:hasNorm ?EnvironmentNorm)]
```

3 CASE STUDY

The Supply Chain Management (SCM) game from the Trading Agent Competition (TAC) is the case study chosen to be presented in this work. The TAC is an international forum designed to promote and encourage high quality research into the trading agent problem. The TAC SCM game [5] was designed to capture many of the challenges involved in supporting dynamic supply chain practices, while keeping the rules of the game simple enough to entice a large number of competitors to submit entries.

A supply chain consists of all parties, directly or indirectly, that have the main goal of fulfilling customer requests [12]. The planning and coordination of activities in a supply chain is the main concern of the supply chain management. Nowadays, an effective supply chain management is vital to obtain competitiveness in constantly changing markets. Current supply chains are still static and rely on long-term relationships among key trading partners. However, there is a need for more flexible and dynamic practices that can offer better matches between suppliers and customers as market conditions change.

For the case study, some laws from the TAC SCM specifications [13] were identified and classified according to the four levels of regulations proposed in this work. The classifications were based on the definitions of the levels of regulations given in the Section 2, but it is known that the border between each level, sometimes, is unclear. For those cases, the classifications can be discussed. To simplify the presentation, just one environment (the TAC environment), one organization (the TAC main organization) and three roles (supplier, manufacturer and customer) will be considered.

1. Environment Laws:

1.1. Trading designed to benefit some other agent at the expense of the trader's own utility is not allowed.

1.2. Denial-of-service attacks are not allowed. Agents may not employ API operations for the purpose of occupying or loading the game servers.

1.3. Organizations cannot be composed of more than six agents playing together.

1.4. Organizations have a fixed life time of 220 days.

1.5. To get in the environment, agents have to connect to a game server.

1.6. Communication with the agent during a game is not allowed. Agents may obtain runtime game information only via the defined auction API.

2. Organization Laws:

2.1. Is allowed to carry a negative balance in agents' bank account.

2.2. Organizations have to follow the model *direct sales to customer*, i.e., sales in the organization can just be made between the suppliers and manufacturers or between manufacturers and customers.

3. Supplier Role Laws:

3.1. Only complete orders are shipped. The only exception is in the last day of the game when partial orders are shipped.

3.2. Orders are not shipped before their due dates.

3.3. Every order has a down payment of 10%.

4. Manufacturer Role Laws:

4.1. Every shipped order must be paid.

4.2. Every manufacturer can only produce if all the required components are available in his inventory.

4.3. Every manufacturer can only ship products if they are available in his inventory.

4.4. Every manufacturer has an assembly cell that cannot process more than 2000 cycles/day.

5. Customer Role Laws:

5.1. Every shipped order must be paid.

5.2. Every request for quotes (RFQs) has to have specified in it: product type, quantity due date, reserve price, penalty amount and maximum price per unit that the customer is willing to pay.

5.3. The valid bid with the lowest price has to be chosen.

5.4. A randomly choice has to be made when valid bids tied.

6. **Interaction Laws between Manufacturers and Suppliers:**

   6.1. Suppliers have to answer all manufacturers' RFQs.

   6.2. Manufacturers can only send 5 RFQs/day to each supplier for each of the products offered.

   6.3. Suppliers have to ignore subsequent orders if more than one type (*partial offer* or *earliest complete offer*) was done in a day.

7. **Interaction Laws between Manufacturers:**

   7.1. A manufacturer cannot sell to another manufacturer.

8. **Interaction Laws between Manufacturers and Customers:**

   8.1. If a RFQ was sent by a customer to a manufacturer, all others manufacturers have to receive RFQs too.

   8.2. Bid of manufacturers must address the entire quantity specified in the RFQ from customers.

   8.3. Bid of manufacturers must be delivered on the due date specified in the RFQ.

   8.4. Bid prices of manufacturers must be below or equal to the reserve price specified by the customer in the RFQ.

The approach of identifying and classifying the laws according to our four levels of regulations is extremely important for implementing systems in a supply chain domain. In the lifecycle of these systems, new laws are included and existing laws are modified to support new business 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 the TAC game level more difficult, interaction and role laws can evolve and environment and organizations laws don't need to.

3.1 Case Study Implementation

After identifying some TAC SCM laws in its specification document and classifying those in all our levels of regulations, the domain independent normative ontology, presented the in Section 2.1 , is extended and instantiated for the TAC SCM domain. The TAC SCM normative ontology represents the formal base for agents to reason about the specified laws.

The case study was implemented inside the Eclipse platform [15] by using the Java programming language [19] and the Jena API [22] as a programmatic environment for OWL and a rule based engine. The Protégé Editor [32] was also used to extended and instantiated the TAC SCM normative ontology. The agents from the case study were implemented by different developers and were treated as black boxes that have access to normative ontologies.

3.2 Ontology Extension

In TAC SCM, the roles supplier, manufacturer and customer have distinct goals and, consequently, execute specific plans and actions to achieve those. Because of their different characteristics, the Supplier, Manufacturer and Customer concepts were created extending the Role concept. The created extended ontology of the TAC SCM domain is illustrated in Figure.8.

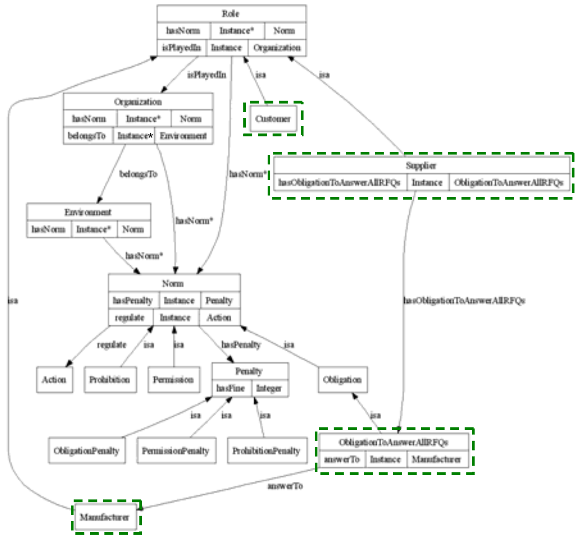

Figure.8. **A normative ontology for a supply chain domain**

The extended ontology has also represented in it all the interaction laws from TAC SCM (laws 6.1 to 8.4), but due to the space limit, Figure.8 just illustrates the law 6.1 as an example. The law is designed by following the representation pattern for interactions presented in Section 2, i.e. the relation object itself is created as a Norm sub-concept and it is related with Role sub-concepts. The related dashed boxes called “Supplier”, “ObligationToAnswerAllRFQs” and “Manufacturer” illustrated in Figure.8 represent the interaction law 6.1.

3.3 Ontology Instantiation

After extending the domain independent normative ontology for the case study, data has to be assigned to it. To simplify the case study, just the “TACEnvironment”, “TACOrganization” and the three roles: “ATACSupplier”, “ATACManufacturer” and “ATACCustomer” were created as instances of the respective concepts: Environment, Organization and Role, from the extended normative ontology. All the identified laws (1.1 to 8.4) from the TAC SCM game are represented as instances of the Permition, Obligation and Prohibition concepts and their associated Penalties are respectively represented as instances of the PermitionPenalty, ObligationPenalty and ProhibitionPenalty concepts. The regulated actions are represented as instances of the Action concept. Due to the space limit, just some instances from the case study are presented below to exemplify its instantiation. All the presented instances are illustrated in Figure.9.

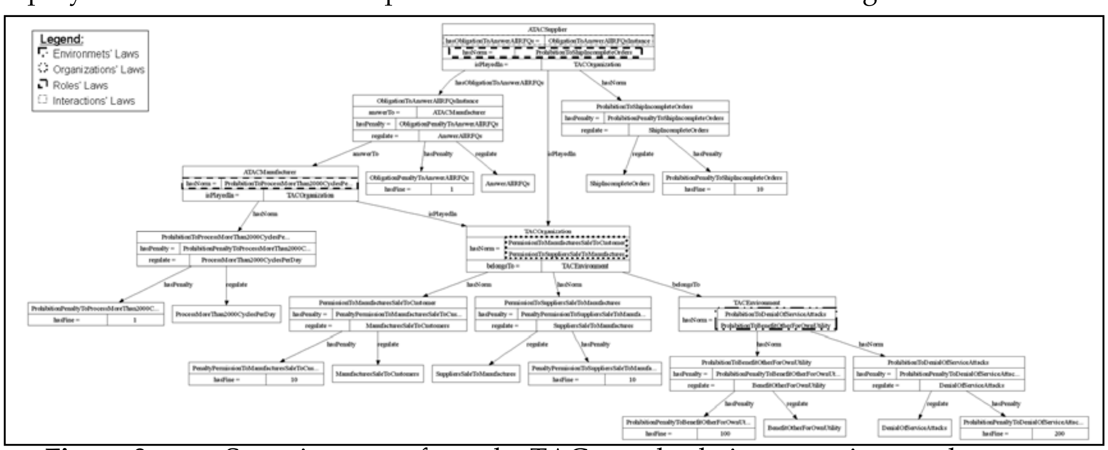

Figure.9. Some instances from the TAC supply chain normative ontology

The “TACEnvironment” environment with its “ProhibitonToBenefitOthersForOwnUtility” and “ProhibitionToDenialOfServiceAttacks” norms represents the environment laws 1.1 and 1.2, respectively. The first norm regulates the “BenefitOthersForOwnUtility” action and has associated to it the “ProhibitonPenaltyToBenefitOthersForOwnUtility” penalty. The second norm regulates the “DenialOfServiceAttacks” action and has associated to it the “ProhibitonPenaltyToDenialOfServiceAttacks” penalty.

The “TACOrganization” organization with its “PermissionToSuppliersSaleToManufacturers” and “PermissionToManufacturersSaleToCustomer” norms represents the organization law 2.2. The first norm regulates the “SuppliersSaleToManufacturers” action and has associated to it the “PermissionPenaltyToSuppliersSaleToManufacturers” penalty. The second norm regulates the “ManufacturersSaleToCustomers” action and has associated to it the “PermissionPenaltyToManufacturersSaleToCustomer” penalty.

The “ATACSupplier” role with its “ProhibitionToShipIncompleteOrders” norm represents the role law 3.1. The norm regulates the “ShipIncompleteOrders” action and has associated to it the “ProhibitionPenaltyToShipIncompleteOrders” penalty.

Interaction laws have also to be instantiated in the TAC SCM ontology. For example, the interaction law 6.1 is represented by the “ATACSupplier” and “ATACManufacturer” roles associated by the “ObligationToAnswerAllRFQsInstance” norm.

4 RELATED WORK

The work presented in this paper (i) is compared to the one presented in [28] (ii). In (ii), regulations take place at the level of interaction laws by managing agents’ interactions in order to achieve higher degrees of predictability. The work (ii) uses the notion of interaction laws as an abstraction to regulate agents' interaction based on a conceptual model for developing laws in open multi-agent systems. This model is composed of static, dynamic and formal definitions. A declarative language – XMLaw – for supporting the conceptual model and a software implementation that allows the enforcement of laws through the interception of agents' interaction are also presented in (ii).

Comparing the works (i) and (ii), two main differences can be assessed: the first main difference is that in (ii), just interaction laws are defined and regulations are based only on this level, while in (i), interaction laws can be combined with environment laws, organization laws and role laws for a more complete regulation; the second main difference is that in (ii), enforcement is done by message interception, when agents are not acting according to the defined interaction laws, while in (i) it is not done. In (i), instead of enforcement, agents are punished by penalties and can be banished from the regulated open MAS, i.e. cannot act inside it anymore. In this way, the messages changed between agents have their privacy kept and the overload from the interception process of all changed messages doesn’t exist.

Our work (i) is also compared to the one presented in [14] (iii). In (iii), is proposed the OMNI (Orgazitional Model for Normative Institutions) framework, which is composed of three dimensions: *Normative*, *Organizational* and *Ontological*, for modeling agent organizations. OMNI has 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 Orgazitional dimensions implemented in a given multi-agent architecture with the mechanisms for role enactment and for norm enforcement.

Comparing the works (i) and (iii), both define a meta-ontology with a taxonomy for regulations in open MAS. In the approaches, norms recommend right and wrong behaviors, which can inspire trust into the regulated MAS. In (iii), enforcement is done by any internal agents from the open MAS, in (i) it is not done. The main difference between the works is that, in (iii) environment laws and role laws are not addressed and in (i) they are.

5 CONCLUSIONS

In this paper we introduced some guidelines for regulations in open MAS with support for agents to reason about laws. In our approach, agents' actions can be regulated by: *environment laws*, *organization laws*, *role laws* and *interaction laws*. The defined laws are represented based on a domain independent normative ontology which has six related main concepts: *Environment*, *Organization*, *Role*, *Norm*, *Penalty* and *Action*. The normative ontology has to be instantiated and also can be extended to be used for specific domains.

A case study from the supply chain domain is presented in this work. For the study, the normative ontology was extended and instantiated based on the laws classified according to our four levels of regulations. As a future work, the layers of logic and rules will be better studied in order to get an efficient data retrieving by inference and to provide a more precise support for agents' reasoning in open MAS.

6 ACKNOWLEDGMENTS

We acknowledge the fund raise from CNPq as part of individual grants and the ESSMA project (552068/2002-0).

REFERENCES

[1] Abdelkader, G.: Requirements for achieving software agents autonomy and defining their responsibility. In Proc. of *Autonomy Workshop* at *AAMAS 2003*.

[2] Agha, G. A.: Abstracting Interaction Patterns: A Programming Paradigm for Open Distributed Systems. In Proc. of *Formal Methods for Open Object-based Distributed Systems*, IFIP Transactions, 1997.

[3] 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.

[4] Artikis, A.; Pitt, J. and Sergot, M.: Animated specifications of computational socities. In Proc. of *AAMAS-2002*, Part III: 1053-1061, Italy.

[5] Arunachalam, R. and Sadeh, N.: *The supply chain trading agent competition*. Electronic Commerce Research and Applications, Volume 4, Issue 1, Pages 66-84, Spring. 2005.

[6] Baader, F.; Calvanes, D.; McGuinnes, D.; Nardi, D. and Patel-Schneider, P. (Eds): *The Description Logics Handbook*. Cambridge University Press, Cambridge, 2003.

[7] Bechhofer, S.; Harmelen, F. van; Hendler, J.; Horrocks, I.; McGuinness, D.L.; Patel-Schneider, P. and Stein, L. A.: *OWL Web Ontology Language Reference*. Section 8.2. Electronic available in <http://www.w3.org/TR/owl-ref/#OWLDL>. Accessed in October, 2005.

[8] Berners-Lee, T.; Hendler, J.; and Lassila, O.: The Semantic Web. *Scientific American*, May, 2001. Electronic available in:

<http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21>. Accessed in October, 2005.

[9] Berners-Lee, T.: *Building the future* slide from the presentation "XML and the Web" occurred in 2000. Electronic available in: <http://www.w3.org/2000/Talks/0906-xmlweb-tbl/slide9-6.html>. Access in October, 2005.

[10] Blumberg, B. M.: *Old Tricks, New Dogs: Ethology and Interactive Creatures*. PhD Thesis from the school of architecture and planning. Massachusetts Institute of Technology. 1997.

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

[12] Chopra, S.; Meindl, P.: *Supply Chain Management – Strategy, Planning, and Operations*. Pearson Prentice Hall, Second Edition, ISBN 0-13-101028-X, 2004.

[13] Collins, J.; Arunachalam, R.; Sadeh, N.; Eriksson, J.; Finne, N.; Janson, S.: *The Supply Chain Management Game for the 2005 Trading Agent Competition*. Full specification of the TAC SCM game. Electronic available in <http://www.sics.se/tac/tac05scmspec_v157.pdf>. Accessed in October, 2005.

[14] 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.

[15] Eclipse Foundation: *Eclipse – an extensible development platform and application frameworks for building software*. Version 3.0. Electronic available in <http://www.eclipse.org/>. Accessed in October, 2005.

[16] Felicíssimo, C. H.; Lucena, C.; Carvalho, G. and Paes, R.: Normative Ontologies to Define Regulations Over Roles in Open Multi-Agent Systems. Extended abstract for: *AAAI Fall Symposium on Roles – an interdisciplinary perspective*, 2005, Virginia, USA. To appear.

[17] 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*), from the "19º Simpósio Brasileiro de Engenharia de Software". Brazil, 2005. Electronic available in <http://www.les.inf.puc-rio.br/seas2005/file/cFelicissimo.pdf>. Accessed in October, 2005.

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

[19] Gosling, J.; Joy, B.; Junior, G.L.S.; Bracha, G.: *The Java Language Specification, Second Edition*. 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 October, 2005.

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

[21] 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.

[22] Hewlett-Packard Development Company: *Jena – a Semantic Web Framework for Java*. Version 2.3. Electronic available in

<http://prdownloads.sourceforge.net/jena/Jena-2.3.zip?download>. Accessed in October, 2005.

[23] Huhns, M.N. and Stephens, L.M.: *Multi-Agent Systems and Societies of Agents*. G. Weiss (Eds.), Multi-Agent Systems, ISBN 0-262-23203-0, MIT Press. 1999.

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

[25] McGuinness, D. L. and Harmelen, F. van.: *OWL Web Ontology Language Overview*. Electronic available in <http://www.w3.org/TR/owl-features/>. Accessed in October, 2005.

[26] 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 October, 2005.

[27] Nute, D.: Defeasible logic. In *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 3, pages 353–395. Oxford University Press, 1987. 528 pages.

[28] 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.

[29] Reynolds, D.: *Jena 2 Inference support*. Electronic available in <http://jena.sourceforge.net/inference/index.html>. Accessed in October, 2005.

[30] Silva, V. T. da.: "Uma linguagem de modelagem para sistemas multi-agentes baseada em um framework conceitual para agentes e objetos". [Title in English: *From a conceptual framework for agents and objects to a multi-agent system modeling language*]. Ph.D. Thesis. Port. Presentation: 31/03/04. 252 p. Advisor: Carlos José Pereira de Lucena. From the computer science department of "Pontifícia Universidade Católica do Rio de Janeiro". March, 2004.

[31] 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 October, 2005.

[32] Stanford Medical Informatics at the Stanford University School of Medicine: *Protégé - A free, open source ontology editor and knowledge-base framework*. Version 3.1.1, build 216. Electronic available in <http://protege.stanford.edu/download/release/full/>. Accessed in October, 2005.

[33] Tyrrell, T.: *Computational Mechanisms for Action Selection*. PhD Thesis from University of Edinburgh. 1993

[34] 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, First International Workshop* (*E4MAS, 2004*), USA, 2004. Lecture Notes in Computer Science 3374 Springer 2005, Pages 1-47. ISBN 3-540-24575-8.

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