DynaCROM: An Approach to Implement Regulative Norms in Normative Multiagent Systems

Carolina Felicíssimo[1,2], Jean-Pierre Briot[2,1], Caroline Chopinaud[2], Carlos Lucena[1]

[1] DI/PUC-Rio, Rua Marquês de São Vicente 225, 22453-900, RJ, Brazil
{cfelicissimo, lucena}@inf.puc-rio.br
[2] LIPVI/ParisVI, Avenue du Président Kennedy, 75016, Paris, France
{jean-pierre.briot, caroline.chopinaud}@lip6.fr

Abstract. A major challenge in normative multiagent systems (NMAS) is how norms can be effectively applied to their entities and easily managed. These tasks are arduous because norms are usually written for general purposes, hindering a more precise regulation. In this paper, we go through an operational approach to implement regulative norms in multiagent systems. In our approach, abstract norms are embodied with domain values according to agents' current contexts. Thus, we try to bridge the gap between the theoretical work on norms and practical normative systems providing a solution to implement concrete regulative norms in NMAS. The rationale of the approach is described via a motivating example from the domain of international supply-chains.

Keywords: Regulative norms, contextual classification, regulatory compliance.

1 Introduction

Openness has led to software systems that have no centralized control and that are formed of autonomous entities [20]. Key characteristics of such systems are heterogeneity, conflicting individual goals and limited trust [2]. We assume that multiagent systems (MAS) is an example of open systems in which the actions of heterogeneous, self-interested agents may deviate from the expected behavior in a context.

In order to prevent malicious actions and to build agent trust in a MAS, regulative norms [4] can be used for defining which actions are *permitted*, *obliged* and *prohibited* to be performed by agents so that the system does not reach an undesirable state. Permissions and prohibitions describe positive/negative authorizations whereas obligations describe responsibilities [24]. These three types of norms represent the three fundamental deontic statuses of an act [1] from deontic logic [33].

Deontic logic enables addressing the issue of explicitly and formally define norms and deal with their possible violation [1]. Agents should have their logics and architectures enhanced with deontic logic [23] for being able to take into account the existence of social norms in their decisions (either to follow or violate a norm) and to react to norm violations by other agents [6]. However, the application of norms in MAS is not a straightforward task since heterogeneity and autonomy rule out any assumption concerning the way *third-party* agents are implemented and behave [16].

Normative MAS (NMAS) as an area of research has become a major issue in the MAS field and it can be situated at the intersection of normative systems and MAS. A NMAS is a system that conforms to or is based on norms [3]. NMAS must allow some facility for norm description and evolution, and support agent reasoning about existing norms.

In this paper we present guidelines to operationally implement regulative norms in NMAS by using our approach named DynaCROM (meaning *dynamic contextual regulation information provision in open MAS*) [14]. From the individual agents' perspective, DynaCROM is an information mechanism that makes application agents to become aware of the norms they are bound to in a given moment. In this way, agents are concerned with precise applicable information and, thus, released from knowing in advance all the norms of the MAS in which they will execute. From the system developers' perspective, DynaCROM is a methodology for norm management in MAS that facilitates the tasks of norm design, implementation and integration with regulated agents.

The remainder of this paper is organized as follows. The next section presents our proposal for appliance of regulative norms in NMAS. The use of other types of norms (e.g., constitutive, procedural, conditional [4]) in DynaCROM is left for another paper. The basis of our approach adopts contextual classification in normative systems. Section 3 introduces a contextual normative ontology created for representing declarative specifications of norms. Section 4 explains how DynaCROM composes related contextual norms and how the result is informed to agents for supporting reasoning. Section 5 describes the way that, from declarative regulative norms, we reach operational ones in NMAS. Finally, before concluding and presenting future work, Section 6 positions our work with respect to other approaches.

2 From Abstract to Concrete Norms in Normative MAS

A major challenge in NMAS is how norms can be effectively applied to their agents and easily managed. These tasks are arduous because norms are usually written for general purposes, hindering a more precise regulation.

In [15], the authors of the paper propose to extend the coordination level of a MAS with a normative level so norms can be integrated during the design and execution time of the system. We follow their same proposition but, furthermore, we propose to extend the normative level with, what we call, a *contextual normative level*. In this level, abstract norms are embodied with domain values according to the context wherein they hold. Our proposition for contextual classification of norms follows the ideas first proposed by Dignum in [9] and then refined in [18], however these works mainly address formal issues while our work the practical ones, providing DynaCROM – an implemented solution as a proof-of-concept for our ideas.

In order to illustrate our proposal, Fig. 1 presents the *Coordination*, *Normative* and *Contextual Normative Levels* of a simplistic supply-chain scenario in which activities (illustrated by linked ellipses) are represented in the three layers (connected by dashed arrows). Norms (illustrated by vertical arrows) are applied in the second and last levels, and contextual norms (illustrated by diagonal arrows) in the last one.

For instance, we consider the *Negotiation* activity. This activity summarizes a set of more specific activities performed between customer and seller agents (e.g., a customer asks a seller its price for an order; the seller responds its price, with or without discounts; the customer accepts the seller's price). The activity is linked to the *Payment* one, and both might be translated in the normative level to:

> *A Payment Norm for Effecting a Negotiation*: Negotiations are obliged to be paid by using the national currency of the seller's country.

We consider that this norm is too abstract and vague and thus applied for general purposes. *Implementation Guideline*: abstract norms must be translated to concrete norms in order to make any effect in a regulated system [18]. For example, the abstract payment norm is contextualized as an environment norm and then concretized in the American and Japanese supply-chain domains with the following instantiations:

An Environment Norm for Effecting a Negotiation: Negotiations are obliged to be paid (i) in *USA*, with *American dollars* (*USD*); and (ii) in *Japan*, with *Yen*.

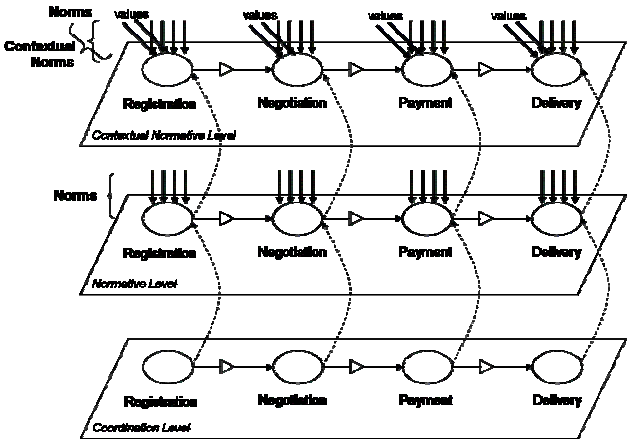

Fig. 1. Activities, Regulated Activities and Contextual Regulated Activities, based on [15].

In the contextual normative level, we follow the classificatory reading of counts-as from [19] stating that if "A counts-as B in context c" then it is interpreted as "A is a sub-concept of B in context c". In this sense, counts-as statements work as *contextual classifications*. For instance, considering the payment norm exemplified above, its reading is done as follows: *USD* counts-as a *valid currency* in the context of the *USA environment*; and its interpretation is as follows: *USD* is a sub-concept of a '*valid currency*' concept in the context of the *USA environment*.

Moreover, besides instantiation of contextualized variables, we consider that activities in the contextual normative level can also have different predefined conditions. For instance, considering a *Give Discount* sub-activity of *Negotiation*, in an organization discounts can be done (a) subtracting 10% of the price value for orders paid in cash, or (b) subtracting 15% of the price value for products bought in bundles.

3 Declarative Specifications of Concrete Norms Using Ontologies

DynaCROM proposes a *contextual normative ontology* for declarative specifications of norms, providing information with a common understanding about well-defined system regulation to heterogeneous agents.

An *ontology* is a conceptual model that embodies shared conceptualizations of a given domain [17]; a *contextual ontology* is an ontology that represents localized domain information [5] (e.g., *USD* is the national currency of *USA*); and a *contextual normative ontology* is a contextual ontology that has a *Norm* concept as a central asset. The *Norm* concept should be instantiated with norms contextualized differently according to basic MAS entities (i.e., environments, organizations, roles and agent interactions [22]) or specific domain entities.

The DynaCROM contextual normative ontology, hereinafter the DynaCROM ontology, defines five concepts: *Role*, *Organization*, *Environment*, *Norm* and *Action*, as illustrated[1] in Fig. 2. These concepts must be instantiated according to the application domain of a NMAS. In this way, the *Role* concept encompasses the instances of all regulated roles of the system; and each role instance has associations with its norms (*hasNorm* property) and organization (*isPlayedIn* property). The *Organization* concept encompasses the instances of all regulated organizations; and each organization instance has associations with its norms, main organization (*hasMainOrganization* property) and environment (*isIn* property). The *Environment* concept encompasses the instances of all regulated environments; and each environment instance has associations with its norms and owner environment (*belongsTo* property). The *Norm* concept encompasses the instances of all norms; and each norm instance has associations with its regulated actions (*regulate* property).

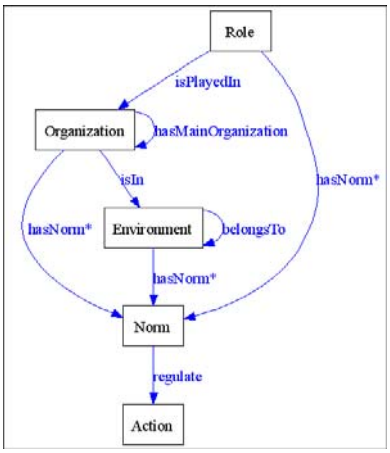

Fig. 2. The DynaCROM ontology.

[1] For readability purposes, ontology is presented graphically by using the Ontoviz graph plug-in [29].

The DynaCROM ontology is an extendible one, i.e., its basic concepts can be extended and/or new domain concepts can be created, both for representing classified contextual domain information. *Implementation Guideline*: the representation of a concrete norm in a DynaCROM ontology should be done by extending existing concepts or by creating new ones, then, instantiating the concept with norm information and, at last, linking the instance to its related abstract norm (represented as a norm instance).

For example, Fig. 3 illustrates the *OblToPayWithNationalCurrency* norm instance that represents an abstract payment norm. This norm is concretized in each environment by the instantiation of the domain data type property *hasCurrency* (e.g., *Yen* in *Japan*, and *USD* in *USA*), which extends the DynaCROM Environment concept (originally presented in Fig. 2).

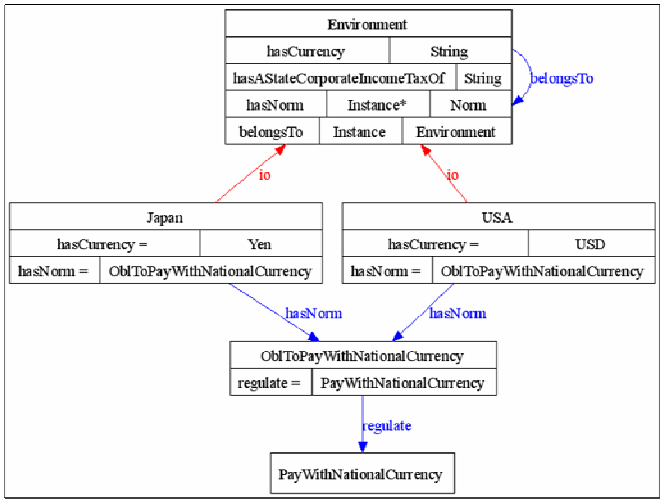

Fig. 3. An abstract environment norm for payments concretized in Japan and USA.

For a more precise regulation in NMAS, domain contexts should also be represented in an application domain DynaCROM ontology through additions and refinements of concepts and their respective norms. An example of an application domain context and its norm can be:

> *A Political Norm for Regulating Deals:* organizations are prohibited from dealing with each other when their countries are undergoing political crisis.

This is an abstract interaction norm. *Implementation Guideline*: interaction norms should be concretized in a DynaCROM ontology by instantiating a *Norm* sub-concept that must be created for linking the other concepts from the relation (i.e. reification of relationship). This solution follows the representation pattern presented in [28].

For example, Fig. 4 illustrates the *PrhToDealWith*, a *Norm* sub-concept, and its instance, both created to concretize the abstract political norm in American and Japanese organizations.

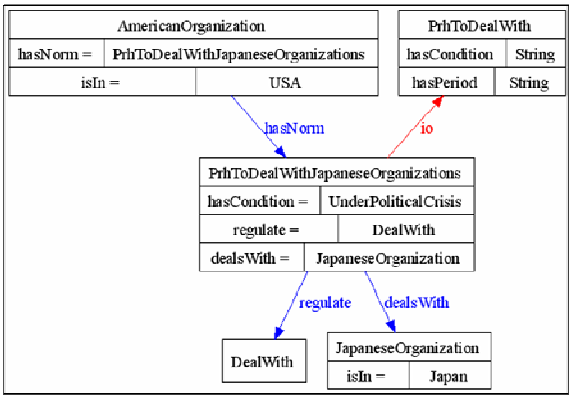

Fig. 4. A political norm concretized in American and Japanese organizations.

4 Contextual Composition and Reasoning in Normative Systems

In order to explain how DynaCROM translates declarative regulative norms (represented in its ontology) to operational information for agent reasoning, we propose the following motivating scenario from a supply-chain domain:

1. An American manufacturer wants to build 100 computers;
2. He issues a call for proposal (CFP) to computer suppliers;
3. Computer suppliers answer the CFP with their proposed price;
4. The manufacturer chooses one proposal among the proposals received and informs his decision to the chosen supplier.

CPU and motherboard are the two main component types necessary to build a computer. Each component type has its base price predefined by suppliers, which are spread through different environments (American states), as illustrated in Table 1.

Table 1. Possibilities for choosing between suppliers of a computer component type.

Component Type	Base price (in USD)	Supplier	State
5.0 GHz CPU	1500	Pintel	California
5.0 GHz CPU	1400	IMD	California
Motherboard	250	Macrostar	Colorado
Motherboard	300	Basus	Missouri

According to the suppliers' locations, the following new environment norms hold:

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

(Concrete) Environment Norms for Calculating Prices: a state corporate income tax rate of (a) *8.84* in *California*; (b) *4.63* in *Colorado*; and (c) *6.25* in *Missouri* is obliged to be imposed on all sales; (d) In *Missouri*, a three day sales tax holiday occurs every year from the first Friday in August until midnight on the Sunday following. Orders of computers and their components, with the maximum cost of $3,500, are eligible for *tax free* status during the holiday season.

Fig. 5 illustrates the environment norms for calculating prices in Missouri. The *OblToImposeAStateCorporateIncomeTax* abstract norm is concretized in each environment by the instantiation of the domain data type property *hasAStateCorporateIncomeTaxOf* (e.g., with the *6.25* value in *Missouri*). This property is created extending the DynaCROM Environment concept. The *PerToNotImposeAStateCorporateIncomeTax* concrete norm was created for representing the period of sales tax holiday. Both norms regulate the calculating price activity that is represented by the *ImposeAStateCorporateIncomeTax* action instance.

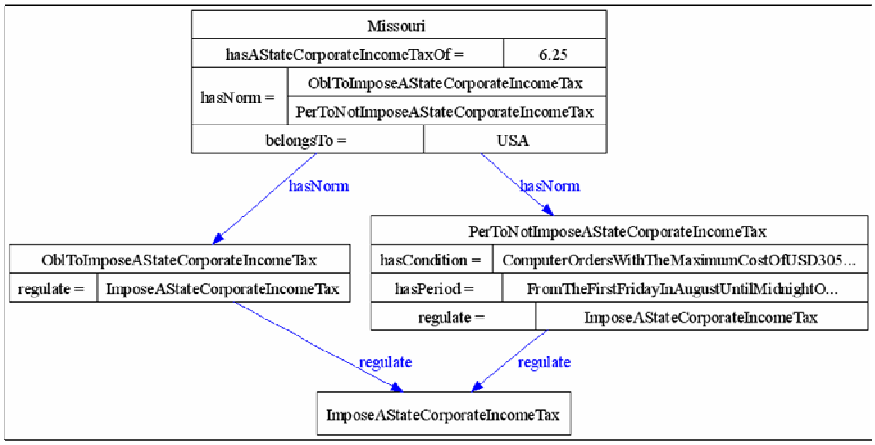

Fig. 5. Environment norms for calculating prices in Missouri.

Although the *Missouri* and *USA* environment instances are linked by the *belongsTo* object property in the DynaCROM domain ontology, a mechanism should be used in order to effectively compose their norms. Different compositions of norms result in sets of independent norms in which the semantic of one norm can influence the semantics of the others.

Implementation Guideline: DynaCROM follows *ontology-driven rules* (i.e., rules created according to an ontology structure) specified by the system developer in order to compose contextual norms. For instance, Code 1 presents an example of rule that recursively compose the norms of hierarchical environments represented in a DynaCROM ontology.

More precisely, considering the *Missouri* environment, the following process is executed: in (4), the '*?OEnv*' variable is instantiated with the *USA* inferred value

when the '*?Env*' variable is instantiated with the *Missouri* given value; in (3), the '*?OEnvNorms*' variable is instantiated with the *OblToPayWithNationalCurrency* inferred value (see Fig. 3); and in (2), the inferred norm is added as a new norm of *Missouri*. The final result of the inference process is that in *Missouri* all negotiations are obliged to be paid in *USD* and, when not during sales tax holiday, increased by a state corporate income tax of *6.25*.

Code 1. A rule to compose the norms of hierarchical environments.

```
(1) [DynaCROMRule_ComposeTheNormsOfHierarchicalEnvs:
(2)   hasNorm(?Env,?OEnvNorms)
(3)    <- hasNorm(?OEnv,?OEnvNorms),
(4)       belongsTo(?Env,?OEnv)]
```

For the norm composition process, DynaCROM has an inference rule engine that executes the following tasks: (i) read a domain ontology instance to get data (i.e., concept instances and their relationships), (ii) read a customized rule file to get how concepts must be composed, and (iii) infer an ontology instance based on the previous readings.

DynaCROM is currently implemented as an active agent behavior[2] for continuously getting agents' updated contexts and their respective norms. All present norms are norms valid in a given moment. Once a domain ontology and/or rule file change, updated norms are automatically forwarded to agents in the next DynaCROM execution. Thus, the dynamics of the process is an important asset of DynaCROM compared with other normative solutions (e.g., [7, 12, 26]) because it permits the management of norms at system runtime and, so, provides the flexibility necessary for regulation regarding social changes characteristic of MAS.

Norm-aware agents are more likely to perform correctly because they are concerned with the applicable information of a regulated system. *Implementation Guideline*: developers should provide to their agents the possibility to add the DynaCROM behavior. Agents enhanced with the DynaCROM behavior are continuously informed about the norms that they are bound to in a given moment according to their current contexts. In this way, agents do not need to have all system norms hard coded inside them, being released from knowing in advance all the norms of each MAS in which they will execute.

For instance, the agents from our example are informed by DynaCROM about the values of state taxes according to their current environments. In this way, agents do not need to have all domain values hard coded inside them and, moreover, they can reason, based on a specific criteria, before choose a supplier. As so, if the American manufacturer's purchase criteria is to minimize costs, then he should choose to buy motherboards in Macrostar because there the state corporate income tax imposed of *4.63* is lower than the one of *6.25* in Basus (both taxes provided by DynaCROM). However, during the three-day sales tax holiday occurring in Missouri, it is better to buy in Basus because no state corporate income tax is imposed during the holiday season (tax information also provided by DynaCROM).

[2] DynaCROM is impplemented in JADE [21], but it can also be applied in other platforms by implementing their facade design pattern provided for the agent unit.

This is a trivial reason for choosing between two organizations, so, we create the following role and interaction norms in the DynaCROM domain ontology instance:

An (Abstract) Role Norm for Accepting Placed Orders: suppliers are obliged to request a down payment for accepting placed orders. *A (Concrete) Role Norm for Accepting Placed Orders*: *Basus suppliers* are obliged to request a down payment of *10%* for accepting placed orders.

An (Abstract) Interaction Norm for Providing Discounts: organizations are permitted to give up to a limited percentage of discounts if their products are bought in bundles. *(Concrete) Interaction Norm for Providing Discounts*: (i) *Basus* and *IMD* organizations are permitted to offer *15%* discount if their products are bought in bundles; and (ii) *Macrostar* and *Pintel* organizations are permitted to offer *10%* discount if their products are bought in bundles.

Fig. 6 illustrates the above role and interaction norms. The *OblToRequestADown-Payment* norm instance represents the abstract role norm for accepting placed orders. This norm is concretized in each role by the instantiation of the domain data type property *requiredDownPaymentOf* (e.g., *10%* in *ABasusSupplier*), which extends the DynaCROM Role concept (originally presented in Fig. 2). The *PerTo-BasusAndIMDSellItemsInBundles* norm instance represents the concrete interaction norm for providing discounts. This norm is concretized by the instantiation of its *giveDiscountOf* and *makeABundleWith* domain data type properties (e.g., *15%* of discount for bundles with *IMD*).

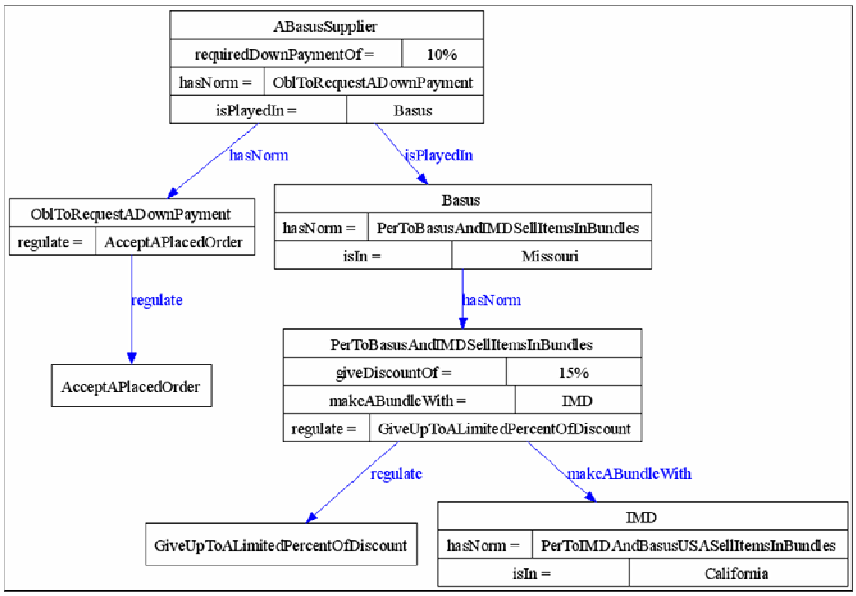

Fig. 6. A role norm for accepting orders and an interaction norm for providing discounts.

According to the present environment, role and interaction norms, now the American manufacturer should choose to buy the bundle IMD/Basus in Basus. Based on the tax information provided by DynaCROM, he can reach the final price for a computer by the following calculation (all in USD): 1,400.00 (IMD CPU) + 300.00 (IMD Motherboard) = 1,700.00 – 15% (IMD/Basus bundle discount) = 1,445.00 + 6.25% (Missouri's state corporate income tax) = 1,535.31. If he decides to buy the bundle in IMD, then the state corporate income tax of 8,84% from California should be applied instead of the 6.25% from Missouri and the final price will be USD 1,566.38. However, the down payment of 10% required by Basus suppliers for accepting placed orders is not anymore necessary.

It is important to remark here that each time a new discount is presented in the DynaCROM domain ontology or even the existing taxes are changed, both managements done for instance by the system developer, updated information is automatically forwarded to the agents enhanced with DynaCROM. Agents without the DynaCROM behavior must implement a similar mechanism for continuously getting system data, according to their current contexts, in order to deal with the applicable norms.

5 From Declarative to Operational Norms for Normative MAS

Norms can be violated respecting agents' autonomy. Although the declarative aspects of norms are important, norms should not only have a declarative meaning but also an operational one. This means that norms should be operationally implemented to be used in practice [31]. Thus, it makes necessary to go from the declarative meaning of norms, which describes how agents should behave in terms of their obligations, permissions and prohibitions, to their operational meaning, which describes the responses to norm violations.

DynaCROM is an approach for implementing dynamic NMAS, in which norms can be updated at system run-time, and also for continuously supporting norm-aware agents with precise information. Notwithstanding, a regulated NMAS need to verify if a performed action is legal or illegal based on its defined norms, which should be enforced. Therefore, experiments were done integrating DynaCROM with SCAAR and MOSES, two solutions for norm enforcement. In SCAAR [7] the norm enforcement is based on agents' internal behaviors; in MOSES [25, 26] it is based on agents' external behaviors. For both solutions, DynaCROM works providing precise norm information as their input.

In the following sub-section we will present the integration of DynaCROM and SCAAR, leaving the one with MOSES for a future publication. Briefly explaining the integration with MOSES, although MOSES is a well-known solution for law enforcement in distributed systems, it only directly enforces interaction norms. Thus, DynaCROM contextual norms have to be decoupled from different levels of abstractions (e.g., environment, organization, role, political levels) to the interaction level. For this, we regard here that communicative acts, established in the interaction level, also can be viewed as organization acts, i.e. actions performed within an organization that modify a fragment of social reality [30]. Moreover, DynaCROM gives dynamics to the MOSES solution in that only the norms applicable to the agents' current con-

texts are sent by DynaCROM to be enforced by MOSES, even if those norms are already defined in a MOSES (*static*) law.

SCAAR

SCAAR (meaning *Self-Controlled Autonomous Agents geneRator*) is a norm enforcement mechanism that enhances agents with a self-monitoring capability for avoiding norm violation. SCAAR utilizes control hooks to trigger an enforcement core each time a regulated action occurs. When norm-aware agents spontaneously incorporate the DynaCROM behavior, aiming to receive updated system norms, they also incorporate SCAAR. In this incorporation process, DynaCROM automatically replaces the headers of the regulated methods implemented inside agents, which have their signature predefined by DynaCROM, to the methods enhanced with the SCAAR control hooks.

Control hooks can be inserted inside agents' code before a regulated action, for preventing norm violation, or after, for detecting norm violation. It is a decision of the system developer when implementing the DynaCROM headers to be replaced in agents' codes. Once a regulated action start running, its control hook triggers the agent enforcement core for the verification and/or enforcement of norm compliance. Norms are represented by Petri nets [27] for verifying compliance, and inhibitor arcs are used for permitting enforcement.

If the system developer decides to use the SCAAR norm prevention mechanism in his regulated MAS then, when a tentative of violation happens with an obligation or prohibition norm, the enforcement core blocks the execution of the infringing action and informs it to DynaCROM; if the system developer decides to only use the SCAAR norm detection mechanism then, when a norm violation happens with an obligation or prohibition norm, the enforcement core informs it to DynaCROM. For a permitted norm, no specific action is taken by SCAAR.

Implementation Guideline: A system developer should write the abstract norms of his regulated system according to both the SCAAR syntax and the DynaCROM domain ontology's structure, and then concretize these norms with instance values in a DynaCROM domain ontology. Regulated actions and norms must be written in the same way in both SCAAR norms and DynaCROM domain ontology. For the enforcement process, DynaCROM reads the ontology and automatically instantiates the abstract norms with domain values, providing concrete norms as input to SCAAR.

SCAAR considers norms written according to the following definition, in which each term represents a set of clauses.

Norm Definition. N (a DO A [AND P]) [IF (a BE in S [AND P])]

```
N := OBLIGED | FORBIDDEN
a := an agent playing a system role
A := an action expression
P := a proposition concerning A or S
S := a state
```

For instance, Code 2 presents a SCAAR norm that the system developer can write to regulate the manufacturer agents of his system when they pay a placed order. A manufacturer agent informs its currency via the '*agtCurrency*' variable. Then,

SCAAR compares this value with the value of the '*domainEnv.hasCurrency*' attribute (line 2), which is automatically instantiated by DynaCROM according to the current agent's environment (e.g., with the "*USD*" value when the agent is in "*Missouri*"). The norm is enforced each time that it is presented in the analyzed environment due to the verification occurred in the lines 3 and 4. Hierarchical environments have also inherited norms enforced due to the norm composition presented in Code 1.

Code 2. A SCAAR norm for manufacturer agents pay a placed order.

```
(1)SCAARNorm_OblToPayWithNationalCurrency:
(2)[OBLIGED (agt DO PayWithNationalCurrency(agtCurrency) AND
              (agtCurrency == domainEnv.hasCurrency))
(3) IF (agt BE in Environment AND (agtEnv == domainEnv)
(4)  AND ("OblToPayWithNationalCurrency" isIn
                                        (domainEnv.hasNorm)))]
```

Code 3 presents an example of a SCAAR norm that the system developer can write to regulate the manufacturer agents of his system when they calculate prices. A manufacturer agent informs its tax via the '*agtStCorpIncTax*' variable. Then, SCAAR compares this value with the value of the '*domainEnv.hasAStateCorporateIncomeTaxOf*' attribute (line 2), which is automatically instantiated by DynaCROM according to the agent's current environment (e.g., with the *6.25* value when the agent is in *Missouri*).

Code 3. A SCAAR norm for manufacturer agents calculate prices.

```
(1)SCAARNorm_OblToImposeAStateCorporateIncomeTax:
(2)[OBLIGED (agt DO ImposeAStateCorporateIncomeTax
                                         (agtStCorpIncTax) AND
  agtStCorpIncTax == domainEnv.hasAStateCorporateIncomeTaxOf)
(3) IF (agt BE in Environment AND (agtEnv == domainEnv)
(4)  AND ("OblToImposeAStateCorporateIncomeTax" isIn
                                        (domainEnv.hasNorm)))]
```

Code 4 presents a last example of a SCAAR norm that the system developer can write to regulate the supplier agents of his system when they accept a placed order. A supplier agent informs its down payment via the '*agtDownPayment* variable. Then, SCAAR compares this value with the value of the '*domainRole.requiredDownPaymentOf*' attribute (line 2), which is automatically instantiated by DynaCROM according to the agent's current role (e.g., with the *10%* value when the agent is playing *ABasusSupplierRole*).

Code 4. A SCAAR norm for supplier agents accept a placed order.

```
(1)SCAARNorm_OblToAcceptAPlacedOrder:
(2)[OBLIGED (agt DO RequestADownPayment(agtDownPayment) AND
        (agtDownPayment == domainRole.requiredDownPaymentOf))
(3) IF (agt BE in Role AND (agtRole == domainRole)
(4)AND ("OblToRequestADownPayment" isIn(domainRole.hasNorm)))]
```

6 Related Work

Electronic Agent-Based Organizations. In [32] the OMNI framework (meaning *Organizational Model for Normative Institutions*) is proposed for modeling agent organizations. OMNI focuses on the organization dimension (structuring the global behavior of the society), on the behavior of the agents from the agent perspective, on agent interactions and on a normative structure that is separate from agents of a MAS.

In order to support the development of both closed systems and open, flexible environments, OMNI presents a rigid specification of its structures defining particular fields for the description of scenes, roles and group of roles. There are no normative aspects further than the ones for organizations, roles, group of roles, agent interactions and agents (i.e., only role, scene and transition norms can be specified). The concept of organization is not explicitly presented; its abstraction is present by listing all institutional roles played by agents (e.g., managers, directors, president, etc.).

Currently, OMNI does not provides a solution for the implementation and integration of its specifications in a given NMAS. DynaCROM can provide a flexible solution for implementing agent organizations by representing the OMNI scenes, roles and group of roles in its ontology. Furthermore, this ontology also can be freely enriched with domain concepts and others particular fields for any concept. The integration of organizational data in the NMAS is transparently occurring when agents incorporate the DynaCROM behavior and then, start receiving domain information.

Electronic Agent-Based Institutions. Electronic institutions [11], or simply EI, are agent-based institutions with their focus on the institutional aspect of organizations. The institutional aspect is further divided into normative aspects (norms that enforce behavior) and dialogical aspects (dialogic interactions). In general terms, EI structure agent interactions by establishing the commitments, obligations and rights of participating agents. EI can be specified and verified by using the ISLANDER [10] graphical tool and they can use the AMELI [12] agent-based middleware as an infrastructure that mediates agents' interactions while enforcing institutional norms. The combination of ISLANDER and AMELI allows to support the design and development of open MAS adopting a social perspective.

The main characteristics of EI are: (i) there are no normative aspects further than the ones for roles, agent interactions and agents; (ii) their specifications are often too society-centric in the sense that it completely fix agent interactions in rigid protocols and interfaces; (iii) all possible interactions among agents have to be defined; and (iv) it is difficult, if not impossible, to describe indirect interactions. This is due to the fact that all interacting activity taking place in an EI is purely dialogic by means of direct communication between the agents.

DynaCROM can be used in AMELI by feeding *governor* agents with precise norm information according to agents' contexts, or it can be used in EI in the place of AMELI for enforcing institutional norms. The main advantage in using DynaCROM as an EI enforcement mechanism is that the great number of messages exchanged between agents and their respective governors, and between governor agents and scene manager agents is minimized. This is because with DynaCROM each regulated agent is already enhanced with an enforcement core responsible for enforcing system norms. Yet, regulated agents supported with DynaCROM in EI are relieved to know in advantage all the norms of the EI in which they will play.

7 Conclusion

A MAS is basically constituted of environments, organizations, roles and agents interacting, suggesting different contexts for regulation in NMAS. However, we agree that these contexts are not targeted to a particular application domain; they rather represent a basic set for a general regulation in a NMAS. For a more precise regulation, this set should be improved through additions and refinements of domain concepts and their respective norms. Thus, the motivating question of our research is how to support system developers in their tasks of appliance and management of domain norms in agents of NMAS and also how to continously inform these agents about the norms they are bound to in a given moment.

Our DynaCROM approach intends to be a straightforward method for smoothly applying and managing regulations in NMAS as well as for enforcing precise contextual norms. DynaCROM is still a work in progress, but we agree that it already has contributions for the domain of NMAS. DynaCROM's main contributions are: (i) a contextual normative ontology to explicitly represent the semantics of classified domain norms in a meaningful way (i.e., with a common understanding) for heterogeneous agents; (ii) a definition of a dynamic norm composition process based on ontology-driven rules; and (iii) a solution for enforcing contextualized norms.

DynaCROM is not tightly coupled with a particular enforcement mechanism. In this paper, we present the first results of the integration of DynaCROM and SCAAR for enforcing contextualized norms of a NMAS from the supply-chain domain. The enforcement was done by SCAAR based on agents' internal behaviors and according to the DynaCROM inputs (norms instantiated according to the agents' contexts). We also started an experiment for a norm enforcement based on agents' external behaviors by using the MOSES enforcement mechanism. For future work, we will continue the experiments with SCAAR and MOSES in order to compare their results. Moreover, we intend to analyze how well-founded inputs from DynaCROM can minimize conflicting norms in NMAS.

References

1. Alberti, M.; Gavanelli, M.; Lamma, E.; Mello, P.; Torroni, P. and Sartor, G.: Mapping Deontic Operators to Abductive Expectations, in: the "Journal of Comput. Math. Organ. Theory", vol. 12, Issue 2-3, 2006, pp. 205–225.
2. Artikis, A., Pitt, J., Sergot, M.: Animated specifications of computational societies, in: Proc. of the AAMAS'02, ACM, 2002, Italy, pp. 1053–1061.
3. Boella, G.; Torre, L. van der; Verhagen, H.: Introduction to normative multiagent systems, in: the "Journal of Comput. Math. Organ. Theory", vol. 12, number 2-3, 2006.
4. Boella, G.; Torre, L. van der: Substantive and Procedural Norms in Normative Multiagent Systems, in: Journal of Applied Logic, 2007, to appear.
5. Bouquet, P., Giunchiglia, F., Harmelen, F.v., Serafini, L., Stuckenschmidt, H.: C-OWL: Contextualizing Ontologies, in: Proc. of the ISWC'03, LNCS 2870, 2003, pp. 164–179.
6. Castelfranchi, C., Dignum, F., Jonker, C.M., Treur, J.: Deliberative Normative Agents: Principles and Architecture, 1999, in: Proc. of the ATAL'99, 1999, pp. 364–378.

7. Chopinaud, C., Seghrouchini, A.E.F., Taillibert, P.: Prevention of harmful behaviors within cognitive and autonomous agents, in: Proc. of the ECAI'06, 2006, Italy, pp. 205–209.
8. Dey, A.: Understanding and using context, in: "Personal and Ubiquitous Computing", 5(1), 2001, pp. 4–7, ISSN: 1617-4909.
9. Dignum, F.: Abstract norms and electronic institutions, in: Proc. of RASTA'02, Bologna, 2002, pp. 93–104.
10. Esteva, M.; Cruz, D. de la; Sierra, C.: ISLANDER: an electronic institutions editor, in: Proc. of the AAMAS'02, ACM, 2002, Italy, pp. 1045–1052.
11. Esteva, M.: Electronic Institutions: from specification to development. IIIA PhD Monography. Vol. 19, 2003.
12. Esteva, M.; Rosell, B.; Rodriguez-Aguilar, J.A.; Arcos, J.Ll.: AMELI: An Agent-based Middleware for Electronic Institutions, In: Proc. of the AAMAS'04, 2004, pp. 236–243.
13. Felicíssimo, C.; Lucena, C.; Briot, J.-P.; and Choren, R: Informing Regulatory Dynamics in Open MAS, in the post proceedings of the COIN@AAMAS'06, LNAI 4386 proceedings, Springer-Verlag volume, pp. 147-162, 2007, ISBN: 978-3-540-74457-3.
14. Felicíssimo, C.; Chopinaud, C., Briot, J-P., Seghrouchni, A.E.F.; Lucena, C.: Contextualizing Normative Open Multi-Agent Systems. In: Proc. of the ACM SAC 2008, to appear.
15. Gaertner, D.; García-Camino, A.; Vasconcelos, W.: Distributed Norm Management in Regulated Multi-Agent Systems, in: Proc. of the AAMAS'07, 2007, pp. 624-631.
16. Grizard, A., Vercouter, L., Stratulat, T., Muller, G.: A peer-to-peer normative system to achieve social order, in: Proc. of the COIN@AAMAS'06, 2006, Japan.
17. Gruber, T. R.: A translation approach to portable ontology specifications, in: "Knowledge Acquisition", 5 (2), 1993, pp. 199–220, ISSN: 1042-8143.
18. Grossi, D.; Dignum, F.: From Abstract to Concrete Norms in Agent Institutions, in: Proc. of the FAABS III, 2004. LNCS 3228, 2005, pp. 12–29, ISBN: 978-3-540-24422-6.
19. Grossi, D.; Meyer, J.-J. Ch.; Dignum, F.: Counts-as: Classification or Constitution? An Answer Using Modal Logic, in: Proc. of the DEON'06, LNAI 4048, pp. 115–130, 2006.
20. Hewitt, C.: Open Information Systems Semantics for Distributed Artificial Intelligence, in: "Artificial Intelligence", vol. 47, Issue 1-3, 1991, pp. 79–106, ISSN: 0004-3702.
21. JADE, 2007, Link: http://jade.tilab.com.
22. Jennings, N. R.: On Agent-Based Software Engineering, in: "Artificial Intelligence", 117 (2) pp. 277-296.
23. Jones, A.J.I., Sergot, M.: On the characterization of law and computer systems: the normative systems perspective, in: Deontic Logic in Computer Science, 1993.
24. Kagal, L.; Finin, T.: Modeling conversation policies using permissions and obligations, in the "Journal of Agents and Multi-Agent Systems" (JAAMAS, 07), 2007, 14: pp. 187–206.
25. Minsky, N.: Law Governed Interaction (LGI): A Distributed Coordination and Control Mechanism (An Introduction, and a Reference Manual), Link: http://www.cs.rutgers.edu/~minsky/papers/manual.pdf.
26. Minsky, N.: MOSES, 2007, Link: http://www.moses.rutgers.edu/.
27. Murata, T.: Petri nets: Properties, analysis and applications, IEEE 77(4), 1989, pp.541–580.
28. Noy, N; Rector, A. (Eds.).: Defining N-ary Relations on the Semantic Web: Use with Individuals, 2007, Link: http://www.w3.org/TR/swbp-n-aryRelations.
29. Ontoviz, Link: http://protege.cim3.net/cgi-bin/wiki.pl?OntoViz.
30. Searle, R.: The construction of social reality. NY Free Press.
31. Vázquez-Salceda, J.; Aldewereld, H.; Dignum, F.: Implementing Norms in Multiagent Systems, in: Multiagent System Technologies, LNAI 3187, 2004, pp. 313–327.
32. Vázquez-Salceda, J.; Dignum, V.; Dignum, F.: Organizing Multiagent Systems, in: the "Journal of Autonomous Agents and Multi-Agent Systems", 11 (3), 2005, pp. 307–360.
33. Wright, G.H.v.: Deontic Logic, in: Mind New Series, vol. 60, no. 237, 1951, pp. 1–15.