

# Enriching a MAS Environment with Institutional Services

Andreia Malucelli<sup>1,2</sup>, Henrique Lopes Cardoso<sup>1</sup>, Eugénio Oliveira<sup>1</sup>

<sup>1</sup> LIACC – NIAD&R, Faculty of Engineering, University of Porto  
R. Dr. Roberto Frias, 4200-465 Porto, Portugal

<sup>2</sup> PUCPR – Pontifical Catholic University of Paraná  
R. Imaculada Conceição, 1155, 80215-901 Curitiba PR, Brazil  
{malu, hlc, eco}@fe.up.pt

**Abstract.** Most environments for multi-agent systems limit themselves to providing message transport and white/yellow page services. While these are generic facilities, in some domains other services are necessary, which may map real-world services provided by institutions. The Electronic Institution concept represents the virtual counterpart of real-world institutions, and one of its benefits is to provide a regulated and trustful environment by enforcing norms and providing specific institutional services. This paper presents some of such institutional services. Ontology-based services are provided to assist agent interaction, making the establishment of business agreements more efficient. After the establishment of an agreement through an appropriate negotiation process, it is necessary to verify the execution of the resulting contract. For this, we introduce an institutional normative environment based on the concept of institutional reality and norms.

## 1 Introduction

Multi-agent systems (MAS) applications include two different kinds of approaches. Some problems require system architectures including cooperative agents developed so as to accomplish an overall goal. On the contrary, in other situations agents may represent independent self-interested entities, with no presupposed cooperation besides mere interaction efforts. While the former types of problems may be addressed through a centralized design, the latter comprise open environments where agents interact and may, through negotiation, establish further cooperation commitments.

Although decentralized and dynamic systems are much more appealing, they must be handled with hybrid approaches. A minimum set of requirements is necessary to allow for heterogeneous and independently developed agents to successfully interact. One way of achieving such a common milieu is by defining communication standards, such as those proposed by FIPA [13], which have been implemented in several agent development platforms, such as JADE [18]. However, in terms of agent's interaction, such frameworks typically limit themselves to providing message transport and white/yellow page services [31].

This paper describes a set of additional services – provided in an *Electronic Institution* (EI) framework – that facilitate agent interaction enabling the establishment of a normative environment.

Our background scenario is the domain of e-business automation, comprising not only the use of information gathering and filtering agents but also the establishment and operation of business relationships. Furthermore, we are interested in the process of Virtual Organization (VO) formation and operation, through which agents representing different business units or enterprises come together to address new market opportunities by combining skills, resources, risks and finances no single partner can alone fulfill [10]. In order to be trustful a VO needs to be regulated by appropriate norms.

One of the key factors influencing the adoption of agent-based approaches in real-world business scenarios is trust. When attempting to automate the creation and operation of business relationships, the behavior of agents must be made predictable, by creating a regulated environment that enforces agents' commitments. The notion of an EI [9, 20] is proposed as a means to provide such a regulated and trustable environment, by enforcing norms and providing specific services.

One of the topics presented in this paper is the ontology-based services, which are important when addressing open environments, that is, situations in which a centralized design is neither possible nor desirable. Such services are also proposed by FIPA, although most MAS platforms seem to ignore them. In our EI environment, such services are provided with the intent of enabling the utilization of negotiation protocols by agents using different domain ontologies.

A protocol for negotiating VO formation can be found in [28]. This protocol is used in an institutional negotiation mediation service. A successful negotiation process must result in an explicit contract that can be monitored. We describe the role of the EI in providing a normative environment that can be used as a means to verify agents' compliance with their established commitments. The EI acts as a trustable third-party providing such a monitoring and enforcement service.

The rest of the paper is organized as follows. Section 2 details ontology-based services, describing the integration of JADE and OWL. Moreover, it presents an ontology interaction protocol based on the contract-net protocol. In section 3 we explore the notion of an institutional normative environment. We present our approach including the representation of institutional reality and norms, which are monitored within the environment. Section 4 concludes the paper.

## **2 Institutional Ontology-based Services**

The Electronic Institution (EI) concept represents the virtual counterpart of real-world institutions, and one of its benefits is to provide institutional services. Besides enforcing norms, institutional services should be provided to assist the coordination efforts between agents using heterogeneous domain ontologies which, representing real-world entities, interact with the aim of establishing business relationships. The ontology-based services proposed and implemented are essential to support agent

interaction (suppliers and customers) as a coordination framework, making the establishment of business agreements more efficient.

We have created a set of services (ontology-based services) embedded in the environment in order to ensure an effective, meaningful negotiation. The provided services are [23]: (i) currency conversion service, (ii) unit measure conversion service, and (iii) matching terms service.

The currency conversion service may be useful in the calculation of prices when agents are dealing with different currencies. Similarly, the unit measure conversion service may be useful when agents are dealing with different measure units. The currency conversion and unit measures conversion services are provided as web services. The user may choose the preferences for the currency and unit measures (International System of Units (SI), UK Imperial System or US System) in which he prefers to negotiate.

The matching terms service is required when some of the agents does not understand the content of a message, e.g. an item (product/service) under negotiation. This service is implemented based on lexical and semantic similarity measures. We have integrated three different similarities matching, which are: (i) calculating an n-grams [8] value for the attributes and relations of the concepts; (ii) calculating an n-grams value for the description of the concepts, and (iii) applying the LCH method [3] based on WordNet [25] to detect semantic similarity between both concepts.

Afterwards, if at least two of the three methods deliver a result, a final result is calculated using weighting in order to make a statement if the compared concepts (products or services, in the context of this work) have the same meaning. Furthermore, a classification according to the correspondence values, based on an established threshold is done.

Ontology-based services are important to allow negotiation to take place. The mapping between two heterogeneous domain ontologies is done dynamically when an agent requests this service, after a not-understood CFP (Call for Proposal). When agents get their ontology terms matched (i.e., they achieve a common understanding), they identify a business opportunity and may negotiate towards the establishment of a contract.

As agents need to be able to communicate with other agents and perceive the environment, a number of interaction languages, tools and platforms have been developed [17]. It is necessary to be aware of the potency as well as the impact of each language, tools and platforms, and select the appropriate form according to the requirements of the problem domain.

In [31], the authors argue that some popular frameworks such as JADE reduce the environment to a message transport system or broker infrastructure, and even in the FIPA specifications it is hard to find any functionality for the environment beyond message transport or broker systems. The objective would be not to restrict interaction to inter-agent communication because it neglects a rich potential of possibilities for the paradigm of Multi-Agent Systems (MASs).

However, if there exists already an effort to develop platforms, tools and languages, and they have been used successfully in the MAS area, it seems reasonable to integrate and improve them in order to explore the full potential of environments.

To address the problem of how to create agents with heterogeneous ontologies using an automated and integrated approach, we developed a new methodology, since

we were faced with the problem of creating JADE agents with disparate ontologies. First, we created the ontologies using the ontology editor Protégé [16], from Stanford University, and produced a set of OWL (web ontology language) [27] files. Then, we transformed the ontological information into an object-oriented language format suitable for JADE.

If our agents shared a common ontology, it would have been possible to use the JADE's built-in approach. However, in open MAS, as we are considering, the JADE's built-in approach is not applicable, since it does not provide support for the integration of different ontologies.

The implemented platform allows both scenarios. The first one is the case when all agents share the same hard-coded ontology which is by default supported by JADE. It was implemented to explore the JADE's features in order to find out how to use ontologies in JADE. The second case, and the most relevant to work, uses the Jena [19] model interface, which has been used to extract information from the ontologies to implement a transparent mapping mechanism from ontologies to agents. This approach has also the advantage of allowing the use of Protégé as a tool to update the ontologies; otherwise updating an ontology would implicate generating Java sources with the protégé BeanGenerator [2] plug-in and recording portions of the agents source code.

Moreover, the implementation of our negotiation process combines the FIPA Contract Net Interaction Protocol with an additional protocol called Ontology Interaction Protocol (OIP). The former represents the general scenario of agents trading goods or services proposed by FIPA. The latter implements the protocol necessary for solving the interoperability problems, when agents are interacting with the environment requesting the ontology-based services [24].

## **2.1 Combining Ontologies and Agents Technologies**

In order to communicate, agents need to use a common language – a language with an unambiguous syntax, well-defined semantics or meanings and expressive power. Thus, JADE agents exchange ACL (Agent Communication Language) messages that are in a standard and FIPA-compliant form to ensure interoperability by providing a standard set of ACL message structure, and, to provide a well-defined process for maintaining this set [13].

Inside of an agent, ontological information is represented as Java objects, but in the content slot of an ACL message, this ontological information is represented as a string or a sequence of bytes. To achieve translations between these two different types of representation, JADE provides a number of classes structured in several packages, known as the Content Reference Model (CRM) [4]. To create an ontology, the CRM contains the classes: Predicates, Concept and AgentAction, that have to be instantiated. "Predicates" are expressions that say something about the status of the world and evaluate to true or false, "Concepts" are expressions that indicate entities that exist and that agents talk and reason about, and "AgentActions" are expressions that indicate something that can be executed by some agent.

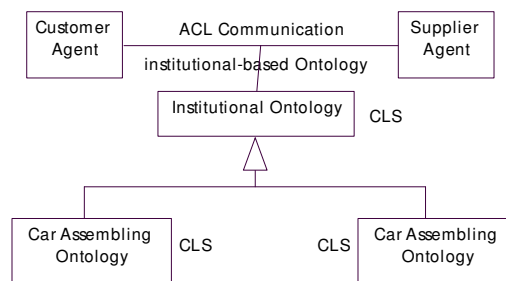
The total of all classes make up our ontology and every single agent involved in a negotiation process has to use these classes. This means that there must be prior

agreement not only about the name used to identify the ontology but as well about sharing the ontological classes. We have compensated this limitation for agents using heterogeneous ontologies creating a shared top-level ontology while each agent has its own private domain ontology.

**Top-level and Domain Ontology.** We profit from JADE's support to build in hierarchies in ontologies as a way to combine ontologies, thus facilitating code re-usage. We have created two types of ontologies using Protégé: a generic one named *Institutional Ontology* and other ones fitting in the Car Assembling domain named *Car Assembling Ontology* and *Automobile Assembling Ontology*.

*Institutional Ontology* is considered as a top-level ontology, while *Car Assembling Ontology* and *Automobile Assembling Ontology* are considered as a domain ontology. The *Concepts*, *AgentActions* and *Predicates* defined in the top-level ontology describe the basic concepts and relationships invoked when any information in an e-commerce context is expressed in natural language, and therefore, are not only related to the domain of *Car Assembling*. The domain ontology contains the elements a car consists of.

Figure 1 graphically points out the hierarchy (UML representation) in the ontologies. Every agent who wishes to negotiate with others must be acquainted with the ontological classes (“CLS” in the figure) whose objects are used to fill the content of an ACL message. Both, the Customer Enterprise Agent (CEAg) and the Supplier Enterprise Agent (SEAg) are able to interpret unambiguously the messages exchanged. To indicate this, they register the name of the ontology they use. *Institutional Ontology* and *Car Assembling Ontology* are recognized as Java classes.

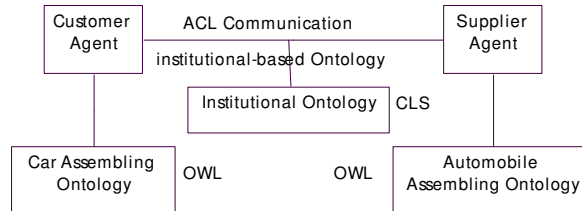


**Fig. 1.** Agents communicating with the same Domain Ontology

However, with agents using heterogeneous ontologies – but designed to describe the same domain of discourse - there are no classes that can be shared. A merging of two or more ontologies might not be an adequate solution in a competitive context since it presumes that every enterprise fully reveals its ontology.

To solve this problem, our approach uses a different way to access the information contained in the ontology of any agent and to search for the required information. The implementation partially abstains from the ontological classes, but choosing another format to provide the knowledge. In this way, the model tries to compensate the

platform's incapability to deal with agents using different ontologies. Figure 2 shows this scenario.



**Fig. 2.** Agents communicating with heterogeneous Domain Ontology

Every agent taking part in a negotiation process shares the *Institutional ontology* and additionally each one explores its own, more specific ontology for the domain of *Car Assembling* (in the example *Car Assembling Ontology* and *Automobile Assembling Ontology*). The *Institutional Ontology* is an ontology that is represented in a suitable way for JADE agents, that is, a set of Java classes. The domain ontologies rely on Web Ontology Language (OWL) files and to handle the OWL files we are using JENA.

Sharing the institutional ontology on the one hand ensures that the agents know exactly the meanings of the messages they are sending and receiving. On the other hand, since agents are able to use their own domain ontology, uniformity is not enforced and consequently the semantic interoperability problem may occur. To solve such problems the ontology-based services are required.

## 2.2 Ontology Interaction Protocol (OIP)

The implementation of a negotiation process combines the FIPA Contract Net Interaction Protocol with an additional protocol called Ontology Interaction Protocol (OIP), as presented in figure 3. The former represents the general scenario of agents trading goods or services proposed by FIPA. Alike other interaction protocols, it structures complex tasks as aggregations of simpler ones. The latter implements the message flow necessary for solving the problems of interoperability, including the interaction of customer and supplier agents when requesting/receiving a service. We have agentified the services with the purpose of facilitating the interaction between the agents and services.

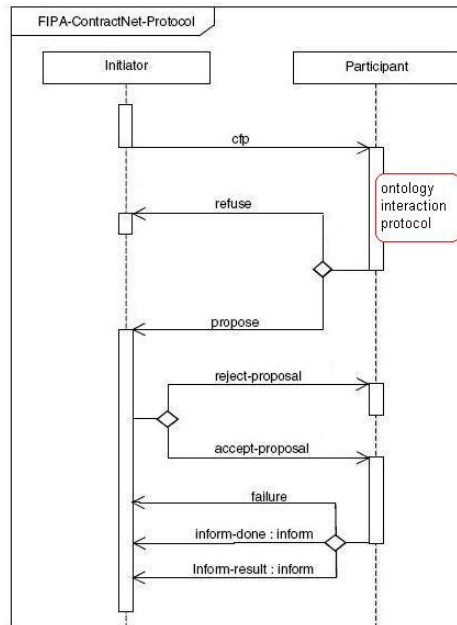
In our scenario, the Customer Enterprise Agent plays the role of the initiator while the Supplier Enterprise Agent is the participant. The initiator wishes to have some task performed and further wants to optimize a function that characterizes the task. This characteristic is commonly expressed as the price, but could also be soonest time to completion, fair distribution of tasks, etc. [14].

For a given task, the participants may respond with a proposal or refuse. Negotiations then continue only with the participants that proposed. The initiator selects among all proposals the best one, based on its own criteria defining what

“best” is, and replies, telling if it accepts the proposal or not. In the former case, once the Supplier Enterprise Agent has completed the task, it sends a message to the Customer Enterprise Agent in the form of an INFORM-DONE or a more explanatory version in the form of an INFORM-RESULT. However, if the participant fails to complete the task, a FAILURE message is sent.

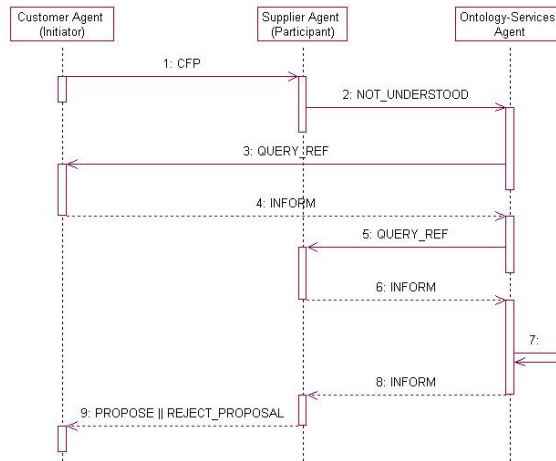
This sequence diagram (figure 3) of the FIPA Contract Net Protocol shows how contracts in general accomplish. JADE provides classes that are implementation of the FIPA Contract Net Protocol.

The agent responding to a CFP (Call for Proposal) performative should answer with a proposition giving its conditions on the performance of the action. The responder's conditions should be compatible with the conditions originally contained in the CFP, e.g., the CFP might seek proposals for an offer for a set of tires, with a condition that the currency is euro. A compatible proposal in reply would be “500 euros for a set of 4 Michelin tires”. An incompatible proposal, for example, would be to use South African rand as currency.



**Fig. 3.** FIPA Contract Net Interaction Protocol and Ontology Interaction Protocol

The sequence diagram in figure 4 represents the implemented ontology interaction protocol (OIP), which intends to find correspondent concepts in two heterogeneous ontologies. The ontology-based services are provided by the Ontology-based Services Agent (OSAg). The Customer Enterprise Agent (CEAg) and the Supplier Enterprise Agent (SEAg) will interact with the OSAg.



**Fig. 4.** Ontology Interaction Protocol

Negotiation proceeds as follow, the numbers in brackets refer to the messages exchanged as depicted in the figure 4.

After having received a CFP (1) as part of the FIPA Contract Net Protocol and not being able to interpret the requested item, the SEAg sends a message with the performative NOT\_UNDERSTOOD to the OSAg (2), acquainting who sent the CFP and the name of the unknown item.

The OSAg sends the name of the item it has just received to the CEAg (3) in order to get further information about it. The CEAg will analyze that request and send back attributes of the concept, their types, price and the description, i.e. all the information about this item (4). The price is taken from its pricelist.

(5) and (6) refer to the pre-selection process. As the name suggests, the pre-selection process aims at getting candidate concepts, which could be the correspondent for the requested product and therefore reducing the target quantity. After having received (4), the OSAg knows the price of the product under negotiation and sends it to the SEAg (5). The process selects among all products the ones whose price value is in the range between 75% and 125% of the received value. This process results in a list of product candidates that is returned to the OSAg, including their names, the characteristics and their description in natural language.

Applying the pre-selection process, we reduce the set of potential matching concepts, which is absolutely essential in huge ontologies defining many entities. Otherwise the number of pairs, meaning concepts that have to be compared, would be too high.

The currency conversion service provided by the OSAg might be needed and can be requested if the SEAg's pricing of items uses a different currency from the requested product. After the selection, the SEAg answers with a list containing names, documentation and attributes of potential correspondent concepts (6).

After receiving all the information about the item under negotiation and a list of possible corresponding items, the OSAg is able to apply methods in order to match the terms (7).



These ontology mapping methods aim at detecting syntactic and semantic similarity of terms. Every term of the proposed, potential correspondent item is compared to the requested term.

In step (8), the OSAg informs the SEAg about the result of the comparisons delivered from the ontology mapping methods, i.e. it informs the name of the correspondent item or an appropriate message if this could not be discovered.

The SEAg is then able to respond to the CEAg (9), either with a PROPOSE or with a REJECT\_PROPOSAL that is part of the FIPA Contract Net Interaction Protocol again.

### **3 Normative Environment**

As exposed before, one of the main aims of the EI is to provide a level of trust through an enforceable normative environment. Norms can play an important role in open artificial agent systems, where they improve coordination and cooperation [6] and allow for the development of trust and reputation mechanisms. As in real-world societies, norms provide us a way to achieve social order [5] by controlling the environment and making it more stable and predictable.

Since we are concerned with the possibility of commitment creation at run-time through the establishment of contracts, our environment has a flexible normative structure. Contractual norms are used to represent agents' commitments. It is the EI responsibility to maintain the normative state of the environment, taking into account the compliance or non-compliance of agents regarding their applicable norms.

Having norms is not sufficient by itself, since agents will not voluntarily submit themselves to associated penalties in case of deviation. Therefore, appropriate mechanisms are needed to enforce norm compliance. The normative environment [22] provides such mechanisms. Therefore, while other institutional services are meant to further facilitate agent interaction (namely the ontology-based services and negotiation mediation), the normative environment comprises an active service that can change the state of the system independently of agents' actions [26]. This is because norm violation can simply be caused by the absence of a certain action, and the occurrence of an associated deadline.

Our EI conceptualization contrasts with other approaches, namely [11], where the EI is seen as a constraining infrastructure implementing a predefined protocol, in which agents are not allowed to violate norms. In our perspective, to enforce norms is not the same as preventing their violation. This approach allows us to maintain the autonomous nature of agents, while influencing their decision making by ensuring that certain consequences will hold in case of non-compliance.

#### **3.1 Institutional Reality**

In order to provide a trustable environment, the EI must have a means to register what is going on. The notion of "institutional reality" embraces the collection of the facts that make it possible to provide norm monitoring services.

**Constitutive Rules.** Part of this institutional reality is achieved by registering events acknowledged by the EI as having occurred. For this we took some inspiration on Searle’s theory on “the construction of social reality” [29]. We distinguish between *brute facts* and *institutional facts*. The latter are obtained from the former, through rules defining “counts-as” relations (*constitutive rules*, according to Searle). Brute facts refer to agents’ illocutions.

Constitutive rules make a connection between what is said and what is taken for granted, by taking into account a set of *institutional roles* enacted by agents providing specific services. Therefore, some institutional facts may come into existence only if agents performing certain institutional roles utter appropriate illocutions. Authoritative relations are thus established between roles and institutional reality: an agent performing a given role is said to be empowered to achieve the effects expressed in its role-related constitutive rules.

For illustration purposes, consider the following simple representation schemes:

- brute facts: `illocution(<Sender>, <Content>)`
- institutional facts: `ifact(<IFact>, <Timestamp>)`
- roles: `agentRole(<Agent>, <Role>)`

Constitutive rules are important to allow the recognition of action execution. This includes the fulfillment of contractual obligations through the realization of certain transactions. Consider that we want to certify an action corresponding to a certain payment obligation. Although the debtor agent may claim to have paid its debt, that does not make it the case. We would instead trust an agent providing a banking service:

```
illocution(?B, payment(?Ag1, ?Amount, ?Ag2, ?Time)) ^
agentRole(?B, bank)
→ ifact(payment(?Ag1, ?Amount, ?Ag2), ?Time)
```

If we need to certify product delivery, we may rely on a delivery tracking service:

```
illocution(?DT, delivery(?Ag1, ?I, ?Qt, ?Ag2, ?Time)) ^
agentRole(?DT, delivery_tracker)
→ ifact(delivery(?Ag1, ?I, ?Qt, ?Ag2), ?Time)
```

If message delivery recognition is a must, a messenger role may provide such a service. This may also be provided as an extension to the message transport service of any agent development platform.

By defining institutional roles instead of institutional agents providing their associated services, we emphasize the open and distributed nature of our institutional environment. Therefore, we may have several agents performing the same institutional role, and thus providing the same institutional service. By “institutional” we mean that those agents are certified by the EI as being trustworthy.

**Institutional Rules.** The purpose of providing an enforceable normative environment must consider other elements, besides institutional facts, that compose the reality to be monitored, and that do not depend directly on brute facts. Taking into account the norms to be monitored, we must consider pending obligations, their fulfillment and violation. The passage of time is also important. All these elements may have interdependencies that may be made explicit by defining *institutional rules*. These rules work on institutional reality elements to produce new elements.

Examples of institutional rules include those that allow us to detect when certain obligations are fulfilled or violated. For illustration purposes, consider the following simple representation schemes for further institutional reality elements:

- obligation: obligation(<Agent>, <IFact>, <Deadline>)
- fulfillment: fulfilled(<Agent>, <IFact>, <Timestamp>)
- violation: violated(<Agent>, <IFact>, <Timestamp>)
- time: time(<Timestamp>)

The <Agent> tag refers to the bearer of the given obligation. We assume that an institutional procedure generates time elements whenever they are relevant for the application of a certain institutional rule.

We may define a rule for verifying the fulfillment of an obligation:

```
ifact(?IFact, ?T) ^
obligation(?Agent, ?IFact, ?Deadline) ^ ?T<?Deadline
→ fulfilled(?Agent, ?IFact, ?T)
```

The rule states that if an institutional fact prescribed by an obligation is achieved before its deadline, then that obligation is fulfilled.

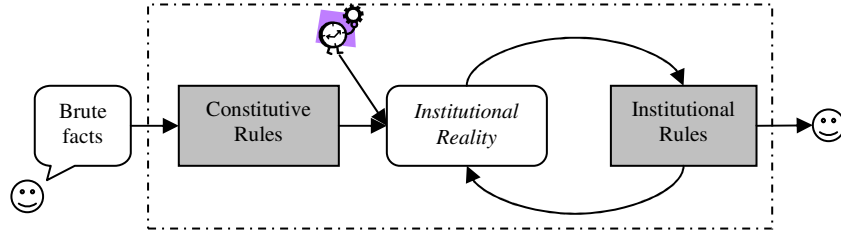
We may also define a rule for detecting the violation of an obligation:

```
ifact(time, ?Deadline) ^
obligation(?Agent, ?IFact, ?Deadline) ^
¬fulfilled(?Agent, ?IFact, ?)
→ violated(?Agent, ?IFact, ?Deadline)
```

The rule states that if a deadline referring to an obligation was reached, and such obligation was not fulfilled, then a violation occurred.

Institutional rules may also point to procedures not amenable to a declarative representation. Examples include rules that trigger notification procedures whenever obligations arise, or rules that impose a reputation update when violations occur.

Figure 5 illustrates the relation between constitutive and institutional rules.



**Fig. 5.** Constitutive and Institutional Rules

This approach to the creation of institutional reality is closely related to the notion of influence and reaction [12]. In our case, influences comprise agents' illocutions (brute facts), through which they try to modify the state of the normative environment, trying to convince the EI that certain events took place. The environment then reacts to such influences by applying the constitutive and institutional rules (the "laws of the world" [12]) and producing institutional facts. However, ours is an asynchronous action model, since agents can run asynchronously and independently of the environment itself (closer to the model in [32]). The next subsection addresses the issue of norms and their relationship with institutional rules.

### 3.2 Norms

A norm-aware environment can operate either preventively (making unwanted behavior impossible) or reactively (detecting violations and reacting accordingly) [30]. In order to cope with the autonomous nature of agents, our approach considers norms as regulations that agents may or may not abide to.

Norms prescribe the expected behavior of agents, specifying states of affairs that *must* be brought about by an agent before a certain deadline. Therefore, we consider *obligations* as the means to express the prescription of behavior norms. Instead of dictating the exact action an agent must perform, obligations prescribe the institutional fact that he must bring about. This fits our model of institutional reality, where we specify by means of constitutive rules how an institutional fact may be accrued.

Just as with institutional rules, norms work on institutional reality elements. The distinguishing feature of norms is that they prescribe obligations when certain conditions are met.

**Example Norms for a Purchase Contract.** A simple set of norms governing a purchase contract is shown below. Contract specific information include the starting time ?S, the vendor ?V, the customer ?C, the item ?I, and the price ?P.

```

→ obligation(?V, delivery(?V, ?I, ?C), ?S+2)
fulfilled(?V, delivery(?V, ?I, 1, ?C), ?TDeliv)
→ obligation(?C, payment(?C, ?P, ?V), ?TDeliv+3)
  
```

```

violated(?V, delivery(?V, ?I, ?C), ?Dln)
→ obligation(?V, delivery(?V, ?I, ?C), ?Dln+5) ^
  obligation(?V, payment(?V, 10%*?P, ?C), ?Dln+5)

violated(?C, payment(?C, ?P, ?V), ?Dln)
→ obligation(?C, payment(?C, ?P*110%, ?V), ?Dln+15)

```

These norms state that the contract starts with a delivery obligation on the vendor, that when fulfilled triggers the obligation on the customer to make the associated payment. The contract also includes two sanctioning norms (based on the violation of obligations), indicating what are the penalties for each case of non-compliance. The institutional rules presented in subsection 3.1 are fundamental for enabling the chaining of obligations within a contractual relationship. They establish a connection between the institutional facts that are added and the pending obligations, verifying their fulfillment or violation, and allowing the applicability of further norms.

**Normative Framework.** Besides the simple norm representation presented above, we consider that a normative environment should be embodied with a set of norms applicable in the absence of further information. An important concept in contract law theory is the use of “default rules” [7], which exist with the intent of facilitating the formation of contracts, allowing them to be underspecified by defining default clauses or default values. The most useful case for this is in defining contrary-to-duty situations (i.e., sanctions), which typically should be not likely to occur. Default regulations provide a normative background in which agents can rely to build their contractual commitments.

Furthermore, taking into account our stated goal of providing assistance to VO formation, we developed a normative framework [21] that considers three hierarchical layers of norms: *institutional*, *constitutional* and *operational*. While institutional norms may be applicable to all agents inside the EI, constitutional norms apply to agents taking part in a VO, and operational norms specify the operationalization of such organizations. Default norms may be defined for each of these layers.

To deal with an environment where a potentially large number of contracts (and VOs) need to be monitored, each of which may include many norms, our formalism considers the use of contextualized norms [22], allowing us to organize them. Also, elements of institutional reality have a context referring to the contract they belong to.

This normative framework, while getting inspiration from law systems in the real-world, comprises a valuable enrichment of MAS environments that need to be regulated.

### 3.3 Implementation

Not surprisingly, our rules and norms are amenable to a rule-based implementation. Also, the normative environment is based on the occurrence of events. We therefore chose a forward-chaining production system as the basis for implementation.

Our normative environment prototype is implemented using the Jess shell [15]. Our knowledge base consists of rules and norms, while the working memory includes institutional reality elements.

Jess also includes the possibility of using frame-based approaches, allowing us to easily aggregate and use contractual information. Jess also has a built-in concept of modules, which we use to organize norms within the system and to employ the default reasoning of the normative environment.

## 4 Conclusions

We have described our approach towards an Electronic Institution that defines a framework for agent activities by adding services to those proposed by agent platforms, building on JADE. The proposed services aim at enriching the MAS environment and point to the creation of organizations of agents through commitments. These are made explicit in contracts that a normative environment is responsible to monitor.

In our open and distributed environment where agents representing different enterprises come together to address new market opportunities, problems as interoperability and trust may happen. In order to help in solving these problems we have presented the ontology-based services and an approach towards the development of a normative environment.

We have implemented a platform integrating agents and ontologies technologies. JADE was used as a communication platform and a taxonomy was applied as a way to combine ontologies to facilitate code re-usage. For the domain ontology creation we used the Ontology editor Protégé with a plug-in, which enables us to store and load our ontology in OWL format. The generic *Institutional ontology* was proposed to ensure that the agents know exactly the meanings of the messages they are sending and receiving, although using their own domain ontology.

The adaptation of the negotiation process combines the FIPA Contract Net Protocol with an additional protocol called Ontology Interaction Protocol, which helps on solving the interoperability problems.

A key factor towards the adoption of agent-based approaches in real-world business scenarios is trust. Therefore, assisting the establishment of agreements is the first step in managing a business relationship. The negotiation process must result in a contract that can be enforced by a third-party. We have presented our approach towards the development of a regulated environment that makes explicit the contractual commitments in order to enforce them.

The agent technology roadmap [1], by AgentLink III, identifies as key problem areas the development of infrastructures for open agent communities, as well as the need for trust and reputation mechanisms. Electronic institutions, together with ontologies and related services, address the needed infrastructures. Norms, electronic contracts and their enforcement are pointed out as means to achieve trust in open environments. Our work is motivated by the need to develop services to assist the coordination efforts between agents which, representing different real-world entities, interact with the aim of establishing business relationships.

## Acknowledgements

The work reported in this paper is supported by the FCT (Fundação para a Ciência e a Tecnologia) Project POSC/EIA/57672/2004.

## References

1. AgentLink III, Agent Technology Roadmap: Overview and Consultation Report. <http://www.agentlink.org/roadmap/index.html>, December (2004)
2. Beangenerator Plug-in, <http://acklin.nl/page.php?id=34>, October (2004)
3. Budanitsky, A., Hirst, G.: Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. Proceedings of the Workshop on WordNet and Other Lexical Resources, Second Meeting of the North American Chapter of the Association for Computational Linguistics. Pittsburgh, USA (2001)
4. Caire, G.: Application-defined Content Languages and Ontologies. JADE Tutorial. TILab S.p.A.. June (2002)
5. Castelfranchi, C.: Engineering Social Order. In: Omicini, A., Tolksdorf, R., Zambonelli, F. (eds.): Engineering Societies in the Agents World Springer (2000) 1-18
6. Conte, R., Falcone, R., & Sartor, G.: Introduction: Agents and Norms: How to fill the gap?. Artificial Intelligence and Law, 7(1) (1999) 1-15
7. Craswell, R.: Contract Law: General Theories. In: Bouckaert, B., De Geest, G. (eds.): Encyclopedia of Law and Economics, Volume III: The Regulation of Contracts. Edward Elgar, Cheltenham (2000) 1-24
8. Damashek, M.: Gauging Similarity via N-Grams: Language-independent Sorting, Categorization, and Retrieval of Text, Science 267 (1995) 843-848.
9. Dignum, V., Dignum, F.: Modelling agent societies: co-ordination frameworks and institutions. In: Brazdil, P., Jorge, A. (eds.): Progress in Artificial Intelligence: Knowledge Extraction, Multi-agent Systems, Logic Programming, and Constraint Solving, LNAI 2258 Springer (2001) 191-204
10. Dignum, V., Dignum, F.: Towards an Agent-based Infrastructure to Support Virtual Organizations. In: Camarinha-Matos, L. M. (ed.): Collaborative Business Ecosystems and Virtual Enterprises Kluwer (2002) 363-370
11. Esteva, M., Padget, J., Sierra, C.: Formalizing a language for institutions and norms. In: Meyer, J.-J., Tambe, M. (eds.): Intelligent Agents VIII Springer (2002) 348-366
12. Ferber, J., Muller, J.P.: Influences and Reaction: a Model of Situated Multiagent Systems. 2nd International Conference on Multi-agent Systems, Japan, AAAI Press (1996)
13. FIPA ACL Message Structure Specification. SC00061G, FIPA TC Communication, <http://www.fipa.org/specs/fipa00061/SC00061G.html>, March (2004)
14. FIPA Contract Net Interaction Protocol Specification, SC00029H, 12/03/2002, <http://www.fipa.org/specs/fipa00029/SC00029H.html>, March (2004)
15. Friedman-Hill, E.: Jess in Action. Manning Publications Co. (2003)
16. Gennari, J., Musen, M.A., Ferguson, R.W., Grosso, W.E., Crubézy, M., Eriksson, H., Noy, N.F., Tu, S.W.: The Evolution of Protégé: An Environment for Knowledge-Based Systems Development, Technical Report. SMI Report Number: SMI-2002-0943 (2002)
17. He, M., Jennings, N.R., Leung, H.-F.: On Agent-Mediated Electronic Commerce, IEEE Transactions on Knowledge and Data Engineering, Vol. 15(4) July/August (2003) 985-1003.
18. Java Agent DEvelopment Framework, <http://jade.tilab.com>

19. Jena Semantic Web Framework, <http://www.hpl.hp.com/semweb/jena.htm>, October (2004)
20. Lopes Cardoso, H., Malucelli, A., Rocha, A.P., Oliveira, E.: Institutional Services for Dynamic Virtual Organizations. In: Camarinha-Matos, L. M., Afsarmanesh, H., Ortiz, A. (eds.): Collaborative Networks and Their Breeding Environments – 6th IFIP Working Conference on Virtual Enterprises (PRO-VE'05) Springer. (2005) 521-528
21. Lopes Cardoso, H., Oliveira, E.: Virtual Enterprise Normative Framework within Electronic Institutions. In: Gleizes, M.-P., Omicini, A. & Zambonelli, F. (eds.): Engineering Societies in the Agents World V Springer. (2004) 14-32
22. Lopes Cardoso, H., Oliveira, E. Towards an Institutional Environment using Norms for Contract Performance. In: Pechoucek, M., Petta, P., Varga, L. Z. (eds.): Multi-Agent Systems and Applications IV – 4<sup>th</sup> International Central and Eastern European Conference on Multi-Agent Systems (CEEMAS 2005). Springer (2005) 256-265
23. Malucelli, A., Oliveira, E.: Using Similarity Measures for an Efficient Business Information-Exchange. In: Skowron, A., Barthes, J-P., Jain, L., Sun, R., Marizet-Mahoudeaux, P., Liu, J., Zhong, N. (eds): 2005 IEEE/WIC/ACM International Conference on Intelligent Agent Technology. IEEE Computer Society, Los Alamitos, California (2005) 234-237
24. Malucelli, A., Palzer, D., Oliveira, E.: Ontology-based Services to help solving the heterogeneity problem in e-commerce negotiations. To be published in Journal of Electronic Commerce Research and Applications - Special Issue Electronic data engineering: the next frontier in e-commerce. Vol. 5(3) Elsevier (2006)
25. Miller, G.: WordNet:A Lexical Database for English. Communication of ACM, 38(11) (1995) 39-41
26. Odell, J., Parunak, H.V.D., Fleischer, M., Breuckner, S.: Modeling Agents and their Environment. In: Giunchiglia, F., Odell, J., Weiss, G. (eds.): Agent-Oriented Software Engineering III. Lecture Notes in Computer Science, Vol. 2585. Springer-Verlag, Berlin Heidelberg New York (2002)
27. OWL – Web Ontology Language, <http://www.w3.org/TR/owl-features/>, October (2004)
28. Rocha, A.P., Lopes Cardoso, H., Oliveira, E.: Contributions to an Electronic Institution Supporting Virtual Enterprises' Life Cycle. In: Putnik, G., Cunha, M.M. (eds.): Virtual Enterprise Integration: Technological and Organizational Perspectives Idea Group, Inc. (2005) 229-246
29. Searle, J. R.: The Construction of Social Reality. Free Press, New York. (1995)
30. Vázquez-Salceda, J., Aldewereld, H., Dignum, F.: Implementing norms in multiagent systems. In: Lindemann, G., Denzinger, J., Timm, I. J., Unland, R. (eds.): Multiagent System Technologies, Springer. (2004) 313-327
31. Weyns, D., Holvoet, T.: A Formal Model for Situated Multi-agent Systems. Formal Approaches for Multi-Agent Systems, Special Issue of Fundamenta Informaticae, 63(2) (2004)
32. Weyns, D., Van Dyke Parunak, H., Michel, F., Holvoet, T, Ferber, J.: Environments for Multiagent Systems, State-of-the-art and Research Challenges. In: First International Workshop on Environments for Multiagent Systems, LNAI, Vol. 3374 (2005)