

Assisting and Regulating Virtual Enterprise Interoperability through Contracts

Henrique Lopes Cardoso, Eugénio Oliveira

LIACC – NIAD&R, Faculty of Engineering, University of Porto
R. Dr. Roberto Frias, 4200-465 Porto, Portugal
hlc@ipb.pt, eco@fe.up.pt

Abstract. Virtual Enterprises are a major trend in enterprise interoperability, configuring cooperative settings in which different parties temporarily share their resources towards a common goal. Virtual Enterprises are created in very dynamic environments, and their temporary nature demands for quick set-up phases. In truly open settings, virtual enterprises may comprise business entities whose previous performance record may not be assessed. Contracts formalizing norms of behavior are needed to allow for verification of cooperative business compliance. Contracts also provide a coarse-grained specification of business processes, enough for verifying observance to promised cooperative efforts. In this paper we relate these issues to the increasingly important dichotomy in multi-agent systems: autonomy and openness vs. regulation. We present our approach, including agent-based services provided inside a regulated environment – an *electronic institution*. We then give particular emphasis to contract formalization through norms, their monitoring and enforcement. Our approach assigns contexts to norms, configuring a structured view that addresses the possibility of creating different agents' cooperation agreements inside the overall framework.

1 Introduction

Virtual Enterprises (VE) are a major trend in enterprise interoperability. Some of the key aspects of an evermore dynamic business environment are specialization and dynamism. The VE concept has been applied to many forms of cooperative business, such as outsourcing, supply chains, or temporary consortiums. We are mainly interested in the latter, as it addresses the need for flexible and dynamic alliances between enterprises. Thus, a VE has been defined as “a temporary consortium of autonomous, diverse and possibly geographically dispersed organizations that pool their resources to meet short-term objectives and exploit fast-changing market trends” [4].

We distinguish a VE from a tight integration of two business entities in outsourcing or supply chain configurations. In those cases, information technology approaches are focused on managing inter-organizational workflows, providing a fine-grained engagement between parties that leads to middle- or long-term relationships. We focus on more flexible and dynamic relationships, which are required for small and medium enterprises to prosper in a very competitive market.

In “breeding environments” [1] potential partners are already acquaintances and typically have previous common business experiences. This facilitates the construction of business agreements, as parties can rely on relational contracts [6], which specify continuous relationships that are naturally self-enforceable. Instead of relying on a third-party authority, a relational contract is based on repeated interactions and social norms, representing an informal agreement sustained by the value of future relationships. If, however, we consider truly open environments, the assumption that previous performance records may be assessed no longer holds. Therefore, a VE may comprise business entities that have never worked together in the past.

Generally, formal contracts are preferred when establishing relationships between unknown parties. On the other hand, regular partners rely on implicit relationships, supported by trust and by the threat of withholding business from anyone who has broken a promise in the past. In this paper we mainly address the former case, developing on the use of explicit contractual specifications.

We focus on contracts specifying explicit behavior norms at the business level, used for verification of partners’ compliance to their contractual promises. Furthermore, in the case of contracts formalizing VE relationships, they also describe a coarse-grained specification of the underlying multiparty business process. This allows for verifying partners’ observance to promised cooperation efforts. The actual contract performance may be achieved, e.g., using Service-oriented Architectures, which enable business interoperability integration at the information technology level. Nonetheless, a link must be established between this and the business level, where the agreed-upon commitments are specified.

Some recent discussions within the multi-agent systems research community are closely related to the exposed concerns. The multi-agent paradigm allows us to tackle with important issues in the VE scenario, namely in the formation and operation stages of the VE life-cycle. The usual methodology consists of having agents representing different business entities (e.g. enterprises), which then use negotiation and coordination skills to create and perform cooperative business agreements. While agent theory describes agents as autonomous self-interested entities, preferably interacting in open environments, an important issue arises when attempting to apply agents in real world scenarios: how to ensure cooperative outcomes in scenarios populated with self-interested agents? A possible answer to this problem is to regulate the environment, providing incentives for cooperative behavior through normative constraints [2].

In this paper we present our approach towards an agent-based *electronic institution* providing services for assisting and regulating the construction and operation of VE configurations. We focus on the specification of norms as a means to regulate an open environment, allowing for contracts to be specified in a way amenable to monitoring and enforcement activities. In section 2 we present institutional services assisting the VE formation and operation stages. Section 3 addresses the issue of norms and rules regulating the environment, both from an institutional point of view and for specifying contractual agreements. Section 4 points our directions towards implementing contract monitoring and enforcement services. Section 5 concludes.

2 Institutional Services for Virtual Enterprises

The concept of *electronic institution* (EI) has been proposed to regulate the interaction among software agents. The benefit of an EI resides in its potential to assure legitimacy and security to its members, through the establishment of norms [5]. Besides enforcing norms, institutional services should be provided to assist the coordination efforts between agents which, representing different real-world entities, interact towards the establishment of business relationships.

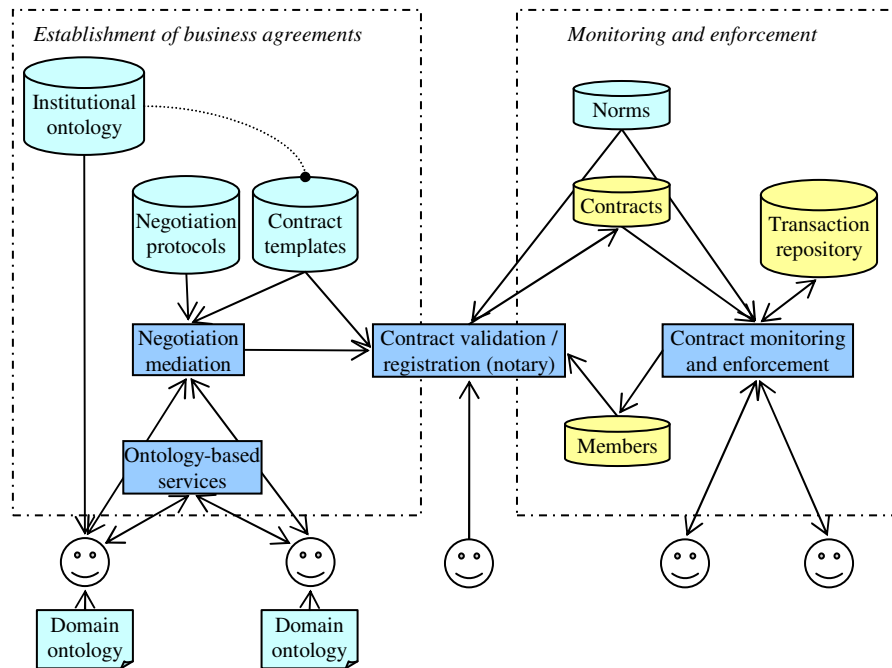


Fig. 1. Services of an Electronic Institution

Our approach regards an EI as a comprehensive framework providing a set of institutional services, while assuring norm enforcement through sanction and reputation mechanisms. As the establishment of business engagements is central to our purposes, we consider an evolving normative environment, including formalizations of “handshakes” by means of contracts that the EI monitors and enforces. We may summarize the main goals of an EI as follows: (1) to support agent interaction as a coordination framework, making the establishment of business agreements more efficient; and (2) to provide a level of trust by offering an enforceable normative environment. These two issues are closely related to the lifecycle of contractual relationships, namely information discovery, contract negotiation and execution. In the particular case of VEs, [12] examines the different stages of their lifecycle, and focuses on advanced negotiation features for the formation stage. In this paper we integrate the formation

and execution stages within the EI environment, establishing a connection between negotiation protocols and the expression of their outcome as contracts.

Considering both goals identified above, we identify agent-based services for contract establishment, monitoring and enforcement (see figure 1, also in [9]).

The establishment of contractual agreements is supported with negotiation mediation, based on appropriate negotiation protocols and contract templates, defined using an institutional ontology. This ontology relates to general contract-related terms. Ontology-based services facilitate the matching process, assisting the interaction of agents using different domain ontologies. These services are important if we aim at keeping an open setting; inside a “breeding environment” common domain ontologies might prevail. The use of templates greatly simplifies the creation of contractual relationships: templates provide a structure on which negotiation can be based, limiting it to the instantiation of template parameters into a mutually agreeable contract.

The validation and registration of contracts allows for their “legal” existence. Contracts are created as a result of a successful negotiation. However, we do not assume that agents will always have their negotiations mediated within the EI. As such, agents may opt to use institutional services for compliance checking only.

An enforceable normative environment is established by rendering a contract monitoring and enforcement service, which registers transactions and verifies norm applicability, as well as the fulfillment of signed contracts.

Every agent intending to use an institutional service must be registered as a member. Agents have, inside the EI’s boundaries, a record of reputation concerning their observance to past contractual relationships. At a later stage of our work, the information in these records will allow for relational contracts to be built.

3 Norms and Contracts

The monitoring and enforcement of contracts is achieved by considering the EI as a normative framework. E-contracts are achieved inside this controlled environment, which establishes certain rules of behavior to be followed by members, ensuring a level of trust that is crucial to the interaction of heterogeneous, independently developed and privately owned agents. The EI imposes a set of institutional norms, not by preventing any deviation from those norms, but instead by ensuring that norm violation is penalized.

Besides mere purchase contracts, we are interested in representing VE relationships by specifying what we call *constitutional norms*. Agents will negotiate to establish dynamic VEs, which stipulate cooperation terms, may exist for a period of time, and are meant to achieve an overall business goal. In order to protect that purpose they must take into account possible misconduct of partners, which may at times have to be replaced. There will typically be a dominant participant, not necessarily for coordinating the whole activity, but because it embodies the final destination of products (as far as the consortium is concerned). This entity can be regarded as a customer (the one that creates a need to be fulfilled by the VE [12]), but can also participate in the production process.

Considering this setting, we conceive a structured normative framework [10] comprising institutional, constitutional and operational norms.

Institutional norms include default contract clauses (or default clause values) that allow concrete contracts to be underspecified, facilitating their creation. General regulations concerning the nature of consortiums may also be defined; this type of policies mirrors existing laws that are created to rule particular aspects of a society. Therefore, agents can rely on these regulations as a ground basis to raise VE contractual formalizations. Rules that recognize violation or fulfillment conditions are also defined at this level, as these are contract-independent concepts. Furthermore, specific policies may be defined towards institutional penalizing of violations (e.g. through reputation mechanisms).

VE constitutional norms describe the terms of cooperation that parties commit to. Our first approach considers that each partner will state workloads and prices for its contribution, and that a general business process outline is specified. This umbrella agreement represents a set of norms that parties adhere to, and which set up the ground for the VE operation phase.

Specific contracts indicating actions to be performed make up the third normative layer. Operational contracts are proposed and signed within the context of VE contractual agreements, and their creation and execution are subject to enforcement and monitoring procedures.

In this section we provide a simple representation of norms, addressing their use in contracts and VE cooperation agreements. We use the Prolog notation conventions for variables and relations, although our rule-based approach is founded on general first-order logic.

3.1 Representing Norms

Following our normative framework, we regard norms as rules prescribing behavior, subject to a certain context. Therefore, a norm is something like:

$$[Context] \textit{Situation} \rightarrow \textit{Prescription}$$

The *Context* indicates the scope of the norm; as such, we may have norms applicable inside the whole institution, while other norms may be defined for a particular VE. The *Situation* describes when the norm is in place. The *Prescription* specifies what should be accomplished in order to the norm to be fulfilled.

Using this representation, norms obviously lend themselves to a rule-based implementation. The situation part refers to the rule conditions, whereas the prescription part corresponds to the rule's effects.

Behavior norms prescribe the expected behavior of agents. Usually, such norms specify 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. Obligations have the following structure:

$$[Context] \textit{obligation}(\textit{Bearer}, \textit{InstitutionalFact}, \textit{Deadline})$$

Instead of dictating the precise action an agent must perform, we prescribe the institutional fact (see below) that he must bring about. This enables an agent to delegate or outsource tasks conducting to the accomplishment of a state of affairs, while still being responsible before the institution for the (un)fulfillment of the obligation.

3.2 Representing Transactions

In our institutional setting transactions are recorded in an appropriate repository, which is used to verify the fulfillment of contractual obligations. We represent such transactions as facts that are acknowledged by the institution. Their representation becomes:

$$[Context] \text{ ifact}(IFact, Timestamp)$$

where *IFact* refers to the reported transaction, and *Timestamp* registers the moment in which such a fact occurred. Just as norms, institutional facts have an associated context. Therefore, a fact denoting the fulfillment of an obligation occurs within the context in which that obligation is in place. For instance, a payment would refer to an invoice issued within a certain contract.

In this paper we refrain ourselves from giving formal details about how institutional facts come about. Briefly, they are inferred from appropriate messages issued by appropriate institutional agents, that is, agents providing specific institutional services, such as banking, delivery tracking, or messaging. These may be rendered as Web services, and permit a certified acknowledgement of real-world transactions. Section 4 briefly addresses this subject.

3.3 Representing Simple Contracts

General contracts for purchase operations between a customer and a vendor can be specified by an institutional fact stating that such a contract is in place, together with a set of behavior norms representing contract clauses that prescribe the expected behavior of contracting parties. A simple example would be:

$$[] \text{ ifact}(\text{purchase_contract}(\text{IdPC}, \text{Customer}, \text{Vendor}, \text{Item}, \text{Quantity}, \text{Price}), \text{PCTime})$$
$$[\text{purchase_contract}:\text{IdPC}] \text{ obligation}(\text{Vendor}, \text{delivery}(_, \text{Item}, \text{Quantity}, \text{Customer}), \text{PCTime}+10)$$
$$[\text{purchase_contract}:\text{IdPC}] \text{ fulfilled}(\text{Vendor}, \text{delivery}(_, \text{Item}, \text{Quantity}, \text{Customer}), \text{TD})$$
$$\rightarrow \text{obligation}(\text{Customer}, \text{payment}(_, \text{Price}, \text{Vendor}), \text{TD}+30)$$

The syntax is simplified for greater readability: in this representation, we assume variable names have contract scope (as opposed to Prolog's clause scope). The absence of a designated agent in institutional facts indicates that such tasks may be delegated (that is, the agent achieving the necessary actions may be other than the obligation's bearer). Within a norm, literals either in the situation or in the prescription part are dependent on the context of the norm.

Sanctioning Norms. Behavior norms may be complemented with sanctioning norms that prescribe new (presumably harder) obligations in case of violation. The general structure of a sanctioning norm is as follows.

$$[Context] \text{ violated}(Bearer, IFact, Deadline) \rightarrow \text{obligation}(Bearer, NewIFact, NewDeadline)$$

This norm states that if the agent identified as the bearer for the achievement of an institutional fact fails to do so by a certain deadline, then a new obligation is assigned to him. Thus, a violation imposes a new obligation on the prevaricator.

3.4 Detecting the Fulfillment and Violation of Obligations

Contextualized institutional facts are used to verify the fulfillment of obligations. For this, we define an obligation *fulfillment rule* applicable to all contexts (that is, to any contractual relationship monitored by the EI):

$$[Context] \text{ ifact}(IFact, T) \wedge \text{obligation}(Bearer, IFact, Deadline) \wedge T < Deadline \\ \rightarrow \text{fulfilled}(Bearer, IFact, T)$$

This rule indicates that if an institutional fact prescribed by an obligation is achieved before its deadline, then that obligation is fulfilled. As with behavior norms, literals within the rule are dependent on its context. That is, if an obligation within a certain contract is accomplished, the fulfillment of such obligation occurs, obviously, inside that same contract. However, this norm is institutional, in that it applies to all contractual relations; it thus has un-instantiated context.

This rule is fundamental for enabling the chaining of obligations within a contractual relationship. It establishes a connection between the institutional facts that are added and the pending obligations.

The connection between behavior norms and sanctioning norms is achieved through violation detection mechanisms. These are based on *violation detection rules*, which fire when deadlines have elapsed. For this we consider time events, which are generated as institutional facts indicating the elapsing of the moments when obligations are due.

$$[Context] \\ \text{ifact}(time, Deadline) \wedge \text{obligation}(Bearer, IFact, Deadline) \wedge \text{not}(\text{fulfilled}(Bearer, IFact, _)) \\ \rightarrow \text{violated}(Bearer, IFact, Deadline)$$

This violation detection rule reads that in any context, if a deadline referring to an obligation was reached, and such obligation was not fulfilled, then a violation occurred (in the same context).

The resulting fact stating that a violation has occurred may be used in several ways: it may affect an agent's reputation; it can constrain the agent's further actions, making it possible to define prohibitions based on violations; it can be used to interdict agent access to institutional services; or it may be used in a less severe way, simply by prescribing new obligations to be attended, as specified in the contract.

This approach allows us to distinguish violation detection from sanction imposition mechanisms. While the detection of violations is a general and institutionally defined concept, the prescription of sanctions may be contract-specific.

3.5 Representing Virtual Enterprise Cooperation Agreements

A cooperation agreement aggregates the organization's constitutional information, including the cooperation effort parties commit to, and their general business process flow. In this subsection we illustrate how to specify this, considering situations where the intended cooperation corresponds to the exchange of resources.

An institutional fact registers that a cooperation agreement is in effect within a group of agents, each ensuring a specific cooperation effort. The following templates are used to specify this information:

[] ifact(cooperation_agreement(IdCA, Participants, Resources), CATime)

*[cooperation_agreement:IdCA]
coop_effort(Participant, Resource, MinQuantity, MaxQuantity, Frequency, UnitPrice)*

[cooperation_agreement:IdCA] business_process(From, Resource, To)

Cooperation efforts indicate quantity ranges for the supply of resources, within a given frequency, together with agreed prices. Business process entries indicate the resources that are supposed to flow between participants. Their effective transfer, however, is dependent on appropriate requests.

A minimum of one *coop_effort* fact per participant and per resource should be present. Also, there should be at least one *business_process* fact for each *coop_effort*, and their agent-resource linkage must conform to the latter. The cooperation agreement contract should also verify that all participants and resources specified within cooperation effort and business process sections are part of the participants and resources declared in the institutional fact. These validations can be performed by the contract validation institutional service.

Cooperation Agreement Intrinsic Norm. The central norm in respect to contractual promises represents the fact that each party is committed to its cooperation effort. As with other behavior norms, this translates to an obligation prescription:

*[cooperation_agreement:IdCA]
ifact(request(Requester, Resource, Quantity, Answerer), TR) ∧
business_process(Answerer, Resource, Requester) ∧
coop_effort(Answerer, Resource, MinQt, MaxQt, Freq, _) ∧
calculate_performed_effort(Answerer, Resource, Freq, TR, PE) ∧ PE+Quantity≤MaxQt
→ obligation(Answerer, acknowledge(Answerer, Resource, Quantity, Requester), TR+10)*

This norm is institutional: it applies to all cooperation agreements created inside the institution. Thus, as before, its context remains unbound until it is in use. Briefly, it reads that if a predicted request (considering the stated business process and coopera-

tion effort) is made in the context of a cooperation agreement, then the envisaged agent is obliged to accept it. An institutional procedure (*calculate_performed_effort*) is invoked for calculating the effort already performed by the agent within the time frame indicated in the cooperation effort frequency, taking into account the request time. If the agent does not exceed its promised efforts, the obligation comes into effect.

Operational Contracts. Operational contracts are established in the context of a cooperation agreement. Institutional facts register their creation:

[cooperation_agreement:IdCA]
ifact(operational_contract(IdOC, Requester, Answerer, Resource, Quantity), OTime)

Considering parties' cooperation commitments, operational contracts come into existence through an institutional rule of the form:

[cooperation_agreement:IdCA]
fulfilled(Answerer, acknowledge(Answerer, Resource, Quantity, Requester), TA)
 \rightarrow *register_new_operational_contract(IdCA, Requester, Answerer, Resource, Quantity, TA)*

This rule applies to every cooperation agreement, and reads that when an agent fulfils its obligation to acknowledge a given request, a new operational contract comes into existence.

The cooperation agreement may also specify how operational contracts are managed, that is, what obligation chains implement such activity. Norms prescribing behaviour on operational contracts may thus be pre-specified when the cooperation agreement is created.

One possibility is to consider operational contracts as purchase contracts using a predefined set of norms:

[cooperation_agreement:IdCA]
ifact(operational_contract(IdOC, Requester, Answerer, Resource, Qt), OTime) \wedge
coop_effort(Answerer, Resource, $_$, $_$, $_$, UnitPrice)
 \rightarrow *ifact(purchase_contract(IdOC, Requester, Answerer, Resource, Qt, UnitPrice*Qt), OTime)*

Another option would be to specify norms applicable to all operational contracts within a cooperation agreement. For instance, if delivery and payment should occur within an operational contract:

[cooperation_agreement:IdCA, operational_contract:IdOC]
obligation(Answerer, delivery($_$, Resource, Quantity, Requester), OTime+10)

[cooperation_agreement:IdCA, operational_contract:IdOC]
fulfilled(Answerer, delivery($_$, Resource, Quantity, Requester), TD) \wedge
coop_effort(Answerer, Resource, $_$, $_$, $_$, UnitPrice)
 \rightarrow *obligation(Requester, payment($_$, UnitPrice*Quantity, Answerer), TD+30)*

in which *IdOC* remains unbound, as these norms apply to all operational contracts which will be created in the future within agreement *IdCA*.

If, however, nothing is pre-specified, then parties must negotiate the terms of operational contracts every time one is created. If they do not, then an institutional default policy may be defined, applicable to all operational contracts of all cooperation agreements.

4 Implementing Contract Monitoring and Enforcement

As mentioned previously, norms lend themselves to a rule-based implementation. Furthermore, the utilization of templates and default norms is amenable to a frame-based development of our normative environment.

For these reasons, we are implementing our agent-based EI framework using the JESS shell [7] for norm representation and enforcement (a central component in our architecture). It incorporates features covering these two issues, besides consisting of a forward-chaining production system that easily fits the application of our norms and rules based on events (in our case institutional facts). JESS also includes the possibility to define modules, which may be used to organize sets of rules. This feature matches our norm contexts.

Our monitoring services rely on a set of institutionally certified agents acknowledging actions related to real-world operations. At a first glance, we identify messaging, banking, and delivery tracking facilities (for information exchanges, monetary value transfers, and product delivery, respectively), which may be provided as Web services and permit a certified acknowledgement of real-world transactions.

The enforcement of contracts is supported by the application of norms triggered by violations, whether those norms simply prescribe new obligations or impose restrictions on agents' actions. Agents' reputation is also affected by their own performance. This information may be later used by other agents when deciding on new contractual experiences.

The effectiveness of contracts demands for the presence of an authoritative third-party (as in [8], [11] or [14]). Moreover, if contract performance is to be monitored, the corresponding transactions must be registered. Another approach could be to delay compliance checking to situations in which conflicts arise and are complaint about. However, this raises problems related to agents' subjective views on contract compliance [3], unless irrefutable evidence is obtained regarding the involved transactions. Having said this, our environment providing the required supervision can be deployed in a decentralized manner, as there may be several institutionally certified agents fulfilling each of the roles associated with the identified services.

5 Conclusions and Related Work

Enterprise interoperability is a growing concern in modern cooperative business. However, the need for flexible and dynamic relationships also demands for open settings, in which potential partners may not have worked together in the past. The specification of contracts offers an alternative when non-legal sanctions (such as reputation

mechanisms) are insufficient to constrain opportunism: parties expose themselves to legal sanctions for non-fulfillment of duties.

Contracts specifying VE configurations have a different nature when compared to simple purchase contracts. They typically define cooperation agreements between several business partners, instead of detailing a predetermined normative path (i.e. a sequence of fixed obligations). A VE relationship has an ongoing nature, and sometimes a flexible structure, where partners may have to be replaced.

Within the framework of our electronic institution providing contract-related services, we have described our structured normative approach, in which norms have attached a context to which they apply. We also have introduced a declarative representation for norms and their use in contracts, together with institutional rules that make a connection between institutional facts and obligations. A simple yet verifiable representation for a type of VE cooperation agreements was also shown, enabling a coarse-grained specification of business processes involving the flow of resources among partners, together with committed cooperation efforts.

The concept of electronic institutions (and the more general topic of normative multi-agent systems) is being addressed by several researchers. Previous approaches towards regulating agent behavior through EIs include [13]. However, this model formally defines an institution using a rigid structure that implements a predefined protocol. It is thus not amenable to contract handling, as contracts typically alter the normative structure. In our approach, we avoid imposing hard constraints on agents' behavior. Through the enforcement of norms, we do influence in some way the behavior of rational agents (in this respect we are much more aligned with [15]). As for norm organization, in [5] a two-level approach is presented, considering norms at institutional and operational levels.

The use of norms for contract specification is widely used (e.g. [8], [14]). Norms are typically related with the deontic notions of obligation, permission and prohibition. Most approaches to norm-based contract specifications deal with these issues, together with associated sanctions in case of violations. In our case, we essentially rely on directed obligations. In the case of VE contracts, permissions are seen as rights for requesting a partner's contribution. Prohibitions are applied as a consequence of violation detection.

The handling of VE contracts is not as much widespread. In [11] a business contract architecture (with some similar components to our approach) can be found (though without a normative perspective). The idea is to provide strong electronic links between organizations that exchange messages carrying business documents, while preserving their individual processes and practices. However, the authors focus on inter-organizational collaborations in an extended enterprise setting (which can be simplistically seen as a VE with only two partners).

Our work addresses issues related to VE interoperability through institutional services provided in an agent-based environment. We therefore put an effort towards a regulated environment – by using concepts such as norms and institutions – that facilitates the employment of agents in a real-world complex problem.

References

1. Camarinha-Matos, L.M., & Afsarmanesh, H. (2003). Elements of a base VE infrastructure. *Journal of Computers in Industry*, 51(2), pp. 139-163.
2. Castelfranchi C. (2000). Engineering Social Order. In A. Omicini, R. Tolksdorf, & F. Zambonelli (eds.), *Engineering Societies in the Agents World*, Springer, pp. 1-18.
3. Daskalopulu, A., Dimitrakos, T., & Maibaum, T. (2001). E-Contract Fulfilment and Agents' Attitudes. In *Proceedings of the ERCIM WG E-Commerce Workshop on the Role of Trust in e-Business*. Zurich, Switzerland.
4. Davulcu, H., Kifer, M., Pokorny, L.R., Ramakrishnan, C.R., Ramakrishnan, I.V., & Dawson, S. (1999). Modeling and Analysis of Interactions in Virtual Enterprises. In *Proceedings of the Ninth International Workshop on Research Issues on Data Engineering: Information Technology for Virtual Enterprises (RIDE 1999)*, IEEE Computer Society, pp.12-18.
5. Dignum, V., & Dignum, F. (2001). "Modelling agent societies: co-ordination frameworks and institutions". In P. Brazdil & A. Jorge (eds.), *Progress in Artificial Intelligence: Knowledge Extraction, Multi-agent Systems, Logic Programming, and Constraint Solving*, LNAI 2258, Springer, pp. 191-204.
6. Hviid, M. (2000), Long-Term Contracts and Relational Contracts. In B. Bouckaert & G. De Geest (eds.), *Encyclopedia of Law and Economics, Volume III: The Regulation of Contracts*, Cheltenham, Edward Elgar, pp. 46-72.
7. JESS, the Rule Engine for the Java Platform: <http://herzberg.ca.sandia.gov/jess/>
8. Kollingbaum, M.J. & Norman, T.J. (2002). Supervised Interaction – Creating a Web of Trust for Contracting Agents in Electronic Environments. In Castelfranchi & Johnson (eds.), *The First International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2002)*, ACM Press, pp. 272-279.
9. Lopes Cardoso, H., Malucelli, A., Rocha, A.P., & Oliveira, E. (2005). Institutional Services for Dynamic Virtual Organizations. In *6th IFIP Working Conference on Virtual Enterprises (PRO-VE'05)*, Valencia, Spain, 26-28 September 2005.
10. Lopes Cardoso, H., Oliveira, E. (2004). Virtual Enterprise Normative Framework within Electronic Institutions. In *Proceedings of the Fifth International Workshop Engineering Societies in the Agents World (ESAW'04)*, Toulouse, France, 20-22 October 2004.
11. Milosevic, Z., Linington, P.F., Gibson, S., Kulkarni, S., & Cole, J. (2004). Inter-organisational collaborations supported by e-contracts. In *I3E: Fourth IFIP Conference on e-Commerce, e-Business, and e-Government*, Toulouse, France, August 2004.
12. Oliveira, E., Rocha, A.P. (2000). Agents Advanced Features for Negotiation in Electronic Commerce and Virtual Organisations Formation Process. In F. Dignum & C. Sierra (eds.), *Agent Mediated Electronic Commerce: The European AgentLink Perspective*, LNAI 1991, Springer, pp. 78-97.
13. Rodríguez-Aguilar, J.A. (2001). *On the design and construction of Agent-mediated Electronic Institutions*, Ph.D. Thesis, Universitat Autònoma de Barcelona.
14. Sallé, M. (2002). Electronic Contract Framework for Contractual Agents. In R. Cohen & B. Spencer (eds.), *Advances in Artificial Intelligence: 15th Conference of the Canadian Society for Computational Studies of Intelligence*, Springer, pp. 349-353.
15. Vázquez-Salceda, J., Aldewereld, H., & Dignum, F. (2004). Implementing norms in multi-agent systems. In G. Lindemann, J. Denzinger, I.J. Timm, & R. Unland (eds.), *Multiagent System Technologies*, LNAI 3187, Springer, pp. 313-327.