A Contract Model for Electronic Institutions

Henrique Lopes Cardoso and Eugénio Oliveira

LIACC – NIAD&R, Faculdade de Engenharia, Universidade do Porto
R. Dr. Roberto Frias, 4200-465 Porto, Portugal
{hlc,eco}@fe.up.pt

Abstract. Electronic institutions are software frameworks integrating normative environments where agents interact to create mutual commitments. Contracts are formalizations of business commitments among a group of agents, and comprise a set of applicable norms. An electronic institution acts as a trusted third-party that monitors contract compliance, by integrating in its normative environment the contractual norms, which are applicable to the set of contractual partners. In this paper we present and explore a contract model that facilitates contract establishment by taking advantage of an institutional normative background. Furthermore, the model is flexible enough to enable the expansion of the underlying normative framework, making it applicable to a wide range of contracting situations.

1 Introduction

Research on norms and multi-agent systems has grown the Electronic Institution (EI) concept as the basis for the development of appropriate normative environments. Such environments are created to establish some kind of social order [4] that allows successful interactions among heterogeneous and autonomous entities.

As with any recent discipline, however, differences exist between the conceptual views of the "institutional environment". Some authors [1] advocate in favor of a restrictive "rules of the game" approach, where the EI fixes what agents are permitted and forbidden to do and under what circumstances. In this case norms are a set of interaction conventions that agents are willing to conform to. Other researchers [2] take a different standpoint, considering the institution as an external entity that ascribes institutional powers and normative positions, while admitting norm violations by prescribing appropriate sanctions. Others still [9] focus on the creation of institutional reality from speech acts, regarding an agent communication language as a set of conventions to act on a fragment of that reality.

A common element in each of these approaches is the *norm*, which enables us to control the environment, making it more stable and predictable. Arguably, one of the main distinguishing factors among researchers using norms in institutions is the level of control one has over agents' autonomy.

Our own view of electronic institutions (as initiated in [14] and developed in [13]) has got two main features that motivate the present paper. Firstly, the institution includes a set of services that are meant to assist (not only regulate) agent interaction and the creation of new normative relationships. This means we do not take the

J.S. Sichman et al. (Eds.): COIN Workshops 2007, LNAI 4870, pp. 27–40, 2008.

environment as static from a normative point of view (as seems to be the case in [1]). New commitments may be established among agents, through contract negotiation (as also noted by [3]); the resulting contracts comprise a set of applicable norms. Additionally, part of the aforementioned assistance is achieved by enriching the institutional environment with a supportive normative framework. This will allow contracts to be underspecified, relying on default norms that compose the institution's normative environment where the contract will be supervised.

In this paper we present and explore the definition of a contract model that takes advantage of an institutional normative framework. The model is flexible enough to encompass contracts of varying degrees of complexity. A contract is established with support of the normative background and relying on a model of institutional reality.

The paper is organized as follows. Section 2 briefly describes the institutional environment supporting the contract model. Section 3 addresses the contract model itself, including its motivation and detailing its constituent parts. The model tries to take advantage of the underlying environment while at the same time enabling the expansion of the normative framework. Section 4 explains contract handling within our electronic institution framework, focusing on the representation of contracts in a computational way. A sample contract is provided for illustration purposes. Finally, section 5 concludes by highlighting the main features of our approach.

2 Institutional Environment

The notion of multi-agent systems assumes the existence of a common environment, where agent interactions take place. Recently more attention is being given to the environment as a first-class entity [17]. In the case of electronic institutions, they provide an environment whose main task is to support governed interaction by maintaining the normative state of the system, embracing the norms applicable to each of the interacting agents.

In order to accomplish such task, in our approach [13] the EI is responsible for recording events that concern *institutional reality*. This reality is partially constructed by attributing institutional semantics to agent interactions.

As mentioned before, we seek to have an EI environment with a supportive normative framework. For this, norms are organized in a hierarchical structure, allowing for norm inheritance as "default rules" [5].

2.1 Elements of Institutional Reality

The institutional environment embraces a set of events composing a reality based on which the normative state of the system is maintained. Norm compliance is monitored consistently with those events, which can be grouped according to their source:

- *Agent-originated events*: in our approach, norm compliance detection is based on the assumption that it is in the best interest of agents to publicize their abidance to commitments. They do so by provoking the achievement of corresponding *institutional facts* (as described in [13]), which represent an institutional recognition of action execution.
- *Environment events*: norms prescribe *obligations* when certain situations arise. In order to monitor norm compliance, the institutional environment applies a set of

rules that obtain certain elements of institutional reality, including the *fulfillment* and *violation* of obligations. While fulfillment acknowledgement is based on institutional facts, violations are detected by keeping track of *time*, using appropriate time ticks. Both norms and rules may use institutional facts as input. Rules also allow obtaining new institutional facts from older ones.

These events are the elements of institutional reality summarized in Table 1.

Table 1. Elements of institutional reality

Element	*Structure*
institutional fact	ifact(<*IFact*>, <*Timestamp*>)
obligation	obligation(<*Agent*>, <*IFact*>, <*Deadline*>)
fulfillment	fulfilled(<*Agent*>, <*IFact*>, <*Timestamp*>)
violation	violated(<*Agent*>, <*IFact*>, <*Timestamp*>)
time	time(<*Timestamp*>)

Because of the normative framework's organization (as explained in the next section), elements of institutional reality are *contextualized*, that is, they report to a certain context defined inside the institutional background.

Our norm definition is equivalent to the notion of conditional obligation with deadline found in [8]. In particular, an *Ifact* (an atomic formula based on a predefined ontology) as included in an obligation comprises a state of affairs that should be brought about, the absence of which is the envisaged agent's responsibility; intuitively, only an achievement of such state of affairs before the deadline fulfills the obligation. The *Deadline* indicates a temporal reference at which an unfulfilled obligation will be considered as violated. Fulfilled or violated obligations will no longer be in effect. Monitoring rules capture these semantics, by defining causal links (as described in [7]) between achievements and fulfillments, and between deadlines and violations.

There is a separation of concerns in norm definition and norm monitoring. The latter is seen as a context-independent activity. Also, the detection of norm (or, strictly speaking, obligation) fulfillment or violation is distinguished from repair measures, which may again be context-dependent (e.g. through contrary-to-duty obligations). This approach differs from [16], where norms include specific violation conditions, detection and repair measures.

2.2 Normative Framework

Our view of the EI concept [13] considers the institution as an environment enforcing a set of institutional norms, but also allowing agents to create mutual commitments by voluntarily adhering to a set of norms that make those commitments explicit. The EI will act as a trusted third-party that receives contracts to be monitored and enforced.

Furthermore, with the intent of facilitating contract formation, we approach the normative framework using a hierarchical approach, enabling the adoption of contract law concepts such as the notion of "default rules" [5]. These enable contracts to be underspecified, relying instead on an established normative background. The grouping of predefined norms through appropriate contexts also mimics the real-world organization of legislations applicable to specific activities. These norms will be imposed when the activity they regulate is adhered to by agents.

Our approach consists of organizing norms through *contexts*. Each contractual relationship is translated into a new context specifying a set of norms while inheriting others from the context within which it is raised. The top-level context is the EI itself.

A context definition includes the information presented in Table 2. The *super-context* (which may often be the EI itself) indicates where the current context may inherit norms from, while the *context type* dictates what kinds of norms are applicable (those that govern this type of relationship).

Table 2. Context definition information

Component	*Description*
super-context	the context within which this context was created
type	the type of context
id	the context identifier
when	the starting date of the underlying contract
who	the participants of the underlying contract

The components described in the table are meant to provide structure to our normative framework. It is the normative environment's responsibility to use this structured context representation in order to find applicable norms in each situation.

The specificity of norms will require further information regarding the contract to which they apply. For this, we consider the explicit separate definition of contextual-information, which will be dependent on the type of context at hand. For instance, in a simple purchase contract, the delivery and payment obligations will need information about who are the vendor and customer, what item is being sold and for what price.

3 Contract Model

This section will provide a description of our proposed contract model. We will start by providing the main assumptions that guided the approach, and proceed with the details of each contract piece. The figures illustrating contract sections were obtained using Altova® XMLSpy®.

3.1 Guidelines

When devising our contract model, we considered the main principles that should guide this definition. On one hand, as stated before we wanted a model that could take advantage of an established normative environment; therefore, each contract should be obtainable with little effort, and with as few information as possible. On the other hand, we also wanted to make the contract model as expansible as possible, allowing for the inclusion of non-predefined information and norms, while still keeping it processable by the EI environment. This requirement will allow us to apply the EI platform to different business domains.

The contract model should therefore allow us to:

– Include information necessary for context creation, and additionally any contract-type-dependent information to be used by institutionally defined norms.

– Add contract-specific details that are meant to override default institutional norms, e.g. by defining contract-specific norms.
– Expand the predicted contract scenarios by enriching the environment's rules for institutional fact generation.

The next sections describe how each of these purposes is handled.

3.2 Contract Header

Although, in general, a contract may include rules and norms, in the extreme case a contract that is to be monitored by the EI may be composed only of its header. Everything else (including the applicable norms) may be inherited from the EI. This minimalist case is illustrated in Figure 1, where dotted lines indicate optional components that we will refer to later. The rounded rectangle with ellipses is a compositor indicating a sequence of components.

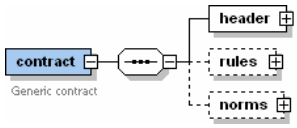

Fig. 1. Generic contract

The contract header (Figure 2) includes mandatory information that is needed for context definition, namely: the contract *id*, the creation date (*when*), and the participants' identification (*who*). The *type* of contract is optional; if not defined, a generic context type will be assumed. The *super-context* is also optional; if omitted, the general EI context is assumed.

Depending on the contract type, some foundational information may need to be provided (e.g. role definitions and goods specification). This information can be included in a frame-based approach: each peace of *contractual-info* (Figure 3) has a name and a set of slots (name/value pairs).

Finally, each contract may indicate the state-of-affairs according to which the contract shall be terminated. The structure of *ending-situation* is analogous to the situation component of a norm definition (as described in the following section).

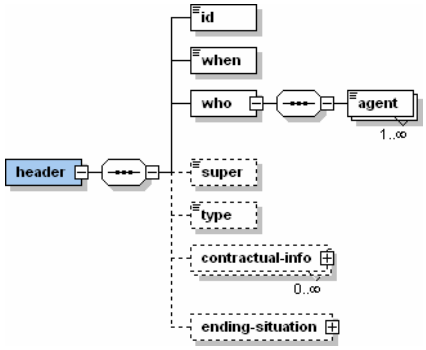

Fig. 2. Contract header

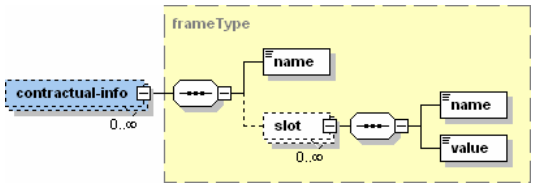

Fig. 3. Contract-type-dependent contractual-info

3.3 Adding Contract-Specific Norms

One way of escaping the default institutional normative setting is by defining norms that are to be applied to a particular contract instance. This is irrelevant of the contract having or not a type as indicated in its heading. A contract of a certain type will inherit institutional norms that are applicable to that type of contract as long as no other contract-specific norms override them. A contract with no type at all will need its norms to be defined in the contract instance.

In our conceptualization, a *norm* prescribes obligation(s) when a certain state-of-affairs is verified (Figure 4). A name is given for norm identification purposes.

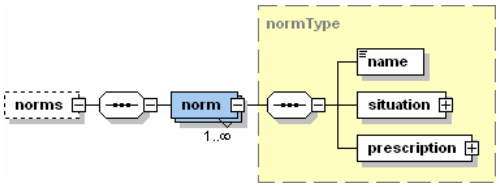

Fig. 4. Contractual norm

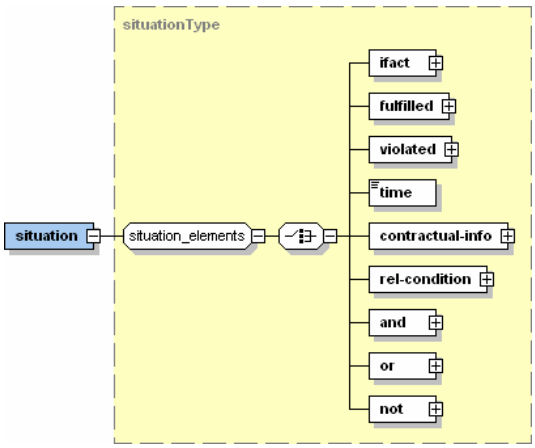

Fig. 5. Situation assessment

The *situation* may be described by institutional reality elements (except obligations) and access contractual-info. Figure 5 includes a choice compositor for situation elements, which may be combined by the logical connectives *and*, *or*, and *not*.

The situation elements *ifact*, *fulfilled* and *violated* match the corresponding institutional reality elements (see Figure 6 and Table 1), as does *time*.

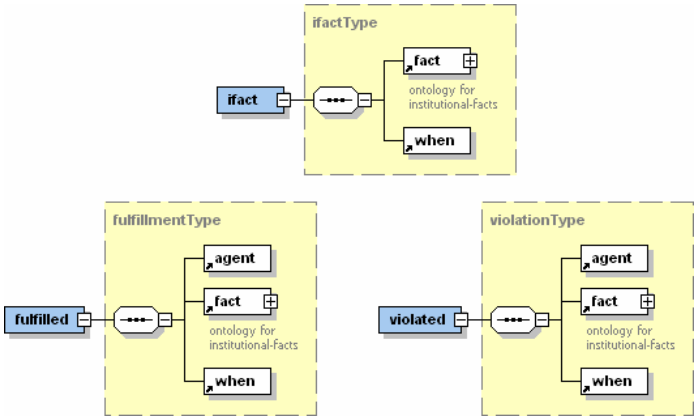

Fig. 6. Situation elements from institutional reality

The inclusion of institutional reality elements and contractual-info inside norms is allowed to use variables for each element's value, such that they can be referred to in other norm components as bounded variables (namely in the prescription part). For that, each element that can hold a variable has an attribute for indicating if the content is a variable name or a value (this approach is adapted from JessML [10]). In order to exploit the institutional ontology, the *fact* element has a frame-like structure similar to that of *contractual-info*. Variables may be used to match slot values inside both of these elements. Restrictions may be imposed through relational conditions that can combine expressions using variables.

The *prescription* of norms includes *obligations* (Figure 7), which have a similar structure to the corresponding institutional reality element. The deadline can be obtained with a numeric expression involving time variables bound in the situation part.

When including norms in a contract-specific way, the normative environment will consider as applicable the most specific norms, that is, those with a narrower scope.

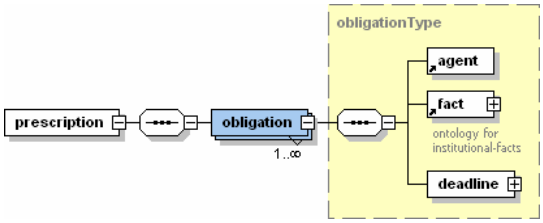

Fig. 7. Obligation prescription

This allows a contract to override predefined norms from a super-context (if specified). The same approach is taken when defining a contract-specific ending situation (in the contract header), which may also be predefined for certain context types.

3.4 Expanding the Creation of Institutional Facts

Following a "counts-as" approach (defining "constitutive rules" [15] or "empowerments" [12]), we attribute institutional semantics to agent illocutions. That is, institutional facts, which are part of institutional reality, are created from these illocutions. This process takes place at an institutional context.

In order to assure the applicability of our environment to different contracting situations, we also included the possibility of iterating through institutional facts (although this is also the case in [15], we take a slightly different perspective [13]). That is, certain contractual situations may consider that certain institutional facts (as recognized by the EI) are sufficient to infer a new institutional fact. The rules that allow these inferences to take place are context-dependent and may be specified in a contract-instance basis (see Figure 8). A rule name is given for identification purposes.

We consider the iterative generation of institutional facts as context-dependent because it allows contract fulfillment to be adjusted by matters of trust between contractual partners or due to business specificities. Thus, it may be the case that only in specific contractual relationships some institutional fact(s) count as another one.

This approach also enhances the expansibility of the system, not restricting norm definition to the institutional fact ontology defined in the preexistent fact-generating rules. It may be the case that a contract defines new institutional facts through these rules and also incorporates norms that make use of them.

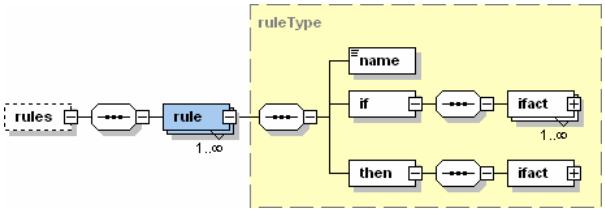

Fig. 8. Rule definition for institutional facts

4 Contract Handling in the Electronic Institution

The contract model described in the previous section comprises an XML schema from which contracts are drafted in the contract negotiation phase. The EI provides a negotiation mediation service for this purpose. After this, the negotiation mediator hands over the contract to a notary service, who collects signatures from the involved agents. After this process is completed, the notary requests the EI to include the contract in its normative environment. The contractual norms will then be part of the normative state of the system, and the normative environment will be responsible for maintaining this state by monitoring the compliance of the involved agents. Figure 9 illustrates this process.

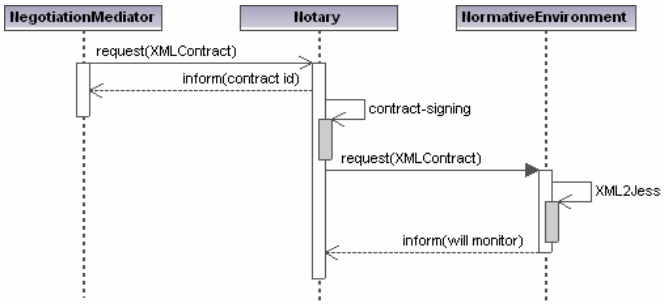

Fig. 9. Contract handling

The figure admittedly underestimates the need for contract validation, which we assume to be implicitly done by the notary and/or the EI. We find this step to be especially relevant when using predefined contract types, which may require the inclusion of foundational information.

As to the contractual norms themselves, in non-electronic practice parties are afforded a considerable degree of freedom in forming contractual relations [6]. Along with this line, our original aim is not to impose predefined regulations on agents, but instead to help them in building contractual relationships by providing a normative background. We therefore do not address for now the issue of predefined norms that are not to be overridden.

4.1 From XML to a Computational Contract Representation

In order to achieve a computational normative environment, a declarative language was chosen for norm representation and processing. Furthermore, in order to facilitate communication with the rest of the agents, the EI includes an agent personifying the normative environment itself. This agent includes an instance of a Jess rule-engine [10], which is responsible for maintaining the normative state of the system and to apply a set of procedures concerning the system's operation.

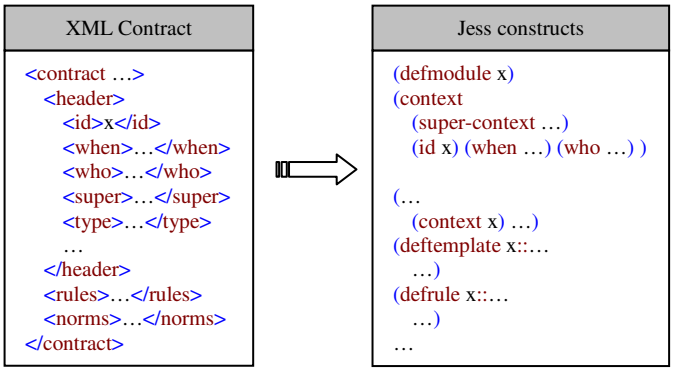

Fig. 10. From XML to Jess

Hence, in order to allow its processing by the normative environment, the XML contract undergoes a process of transformation into appropriate Jess constructs. (see Figure 10). The Jess language includes a set of frame-like constructs.

The generated Jess code will be added to the Jess engine, and comprises information regarding the contract creation (which includes a Jess *module* definition and a context construct), optional contextual-info (and associated Jess *template* definitions), and applicable rules and norms (defined as Jess *rules*).

A rule-based approach to norm representation and monitoring is also pursued in [11]. However, those authors seem to implement in a backward-chaining logic program the semantics of a forward-chaining production system. We follow a more intuitive approach by employing a forward-chaining shell.

4.2 Example

In this section we sketch a simple example of a minimalist contract that illustrates our approach. Figure 11 shows, on the left side, a portion of an XML-contract based on the presented schema. The contract, established by two agents, is a *supply-agreement*; agent *smith* will supply resource *wheel* for a unit price of *10.00*. The right side of Figure 11 shows the resulting Jess code that is generated when adding the contract to the normative environment.

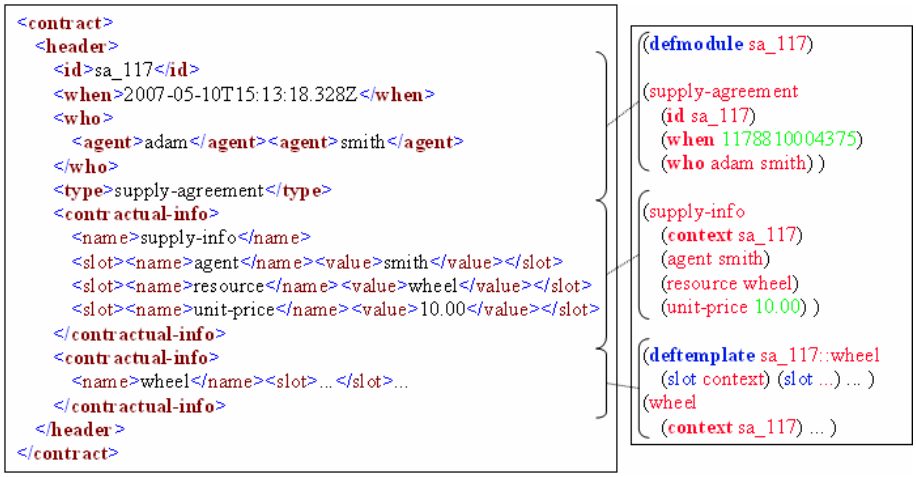

Fig. 11. Sample contract

Taking advantage of the established normative framework, the contract does not specify any norms of its own. It will inherit whatever norms are defined at the normative environment regarding *supply-agreement*s. Figure 12 shows such an applicable norm, together with definitions that make up the normative structure. The upper definitions define the notions of *context* and *contextual-info*; the middle definitions define *supply-agreement* and *supply-info*, which were used in the right side of Figure 11. The lower part of Figure 12 shows a norm applicable to all *supply-agreement*s. Briefly,

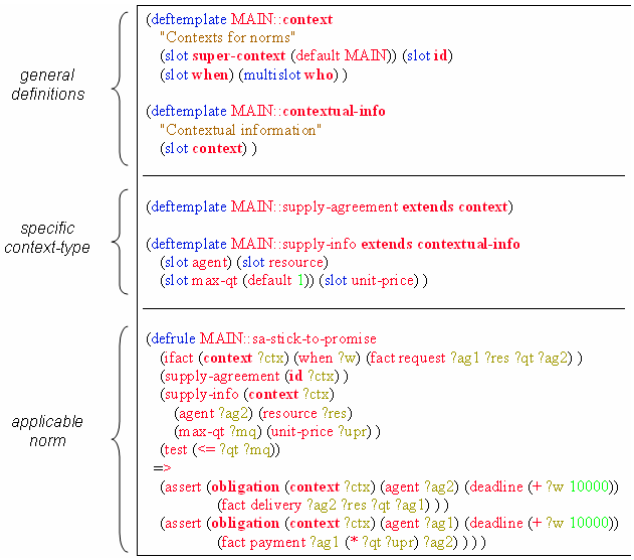

Fig. 12. A predefined norm

that norm states that a request for the furnishing of the promised resource implies an obligation of the supplier to deliver that resource and an obligation of the requester to pay for it.

For lack of space, the example shows only one edge of the spectrum of ways in which the normative environment can be exploited. Contracts can be established that make a partial use of the predefined normative structure, by defining their own specific norms, while still being processable (in terms of monitoring and enforcement activities) by the normative environment. The next section describes the process of norm applicability.

4.3 Norm Monitoring and Inheritance

The module definition and the structured context representation (using *super-context* relations), are the cornerstones for enabling norm inheritance. Norms are defined inside the module representing the contract's context (in the right side of Figure 10, that is what the "*x::*" after *defrule* stands for, where *x* is the module/context name). When applying rules, the Jess engine looks at a focus stack containing modules where to search rules for firing. When no rules are ready to fire in the module at the top of the stack, that module is popped and the next one becomes the focus module.

Exploiting this mechanism, we implemented rules that manage the focus stack and thereby enable the application of the most specific norms in the first place. The event that triggers these rules is the occurrence of a new institutional reality element (IRE), which as explained before pertains to a certain context. Together with the Jess rule engine, our context management rules somewhat implement the algorithm depicted in the flowchart of Figure 13.

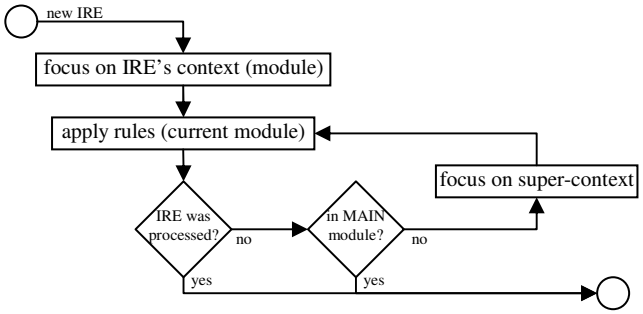

Fig. 13. Processing an institutional reality element

The Jess engine will therefore be guided to look for a module where there is an applicable rule taking the IRE as input. It will start at the IRE's module, and go up one level until the top (*main*) module is reached or the IRE is processed.

This initial exploitation of Jess's features enabled us to start building a proof-of-concept regarding our approach to norm inheritance in a hierarchical normative structure. Further refinements will allow us to configure the system concerning monitoring responsiveness and the integration of social extensions like reputation mechanisms.

5 Conclusions

The EI concept has been approached from different perspectives. Considering the increasing importance of multi-agent system environments [17], the EI can be seen as an interaction-mediation infrastructure maintaining the normative state of the system.

One of the most important principles of our approach is the assumption of a non-static normative environment; this means that we depart from a more conservative view of norms seen as a set of preexistent interaction conventions that agents are willing to comply with (as in the *adscription* approach of [1]). We pursue an EI that provides a supportive normative framework whose main purpose is to facilitate the establishment of further commitments among a group of contracting agents.

The possibility of having an underlying normative framework, from which norms may be inherited, is a distinguishing feature of our approach, as is the "loose coupling" between norms and contrary-to-duties. Also, the institution includes norm monitoring policies that span all created contracts. This is in contrast with other approaches, namely [16], where these policies and repair measures are spread among the norms themselves.

The hierarchical organization of norms takes inspiration in the real-world. The most useful case for "default rules" [5] is in defining contrary-to-duty situations, which typically should be not likely to occur. For this reason, such situations are not dealt with in each contractual agreement, and parties usually recur to law systems that include default procedures [6].

In this paper we presented our approach towards the definition of a contract model that can exploit such an environment. The model was devised taking into account two

aims: it should be easy to compose a new contract, by taking advantage of an institutional normative background; and it should be possible to improve on the EI's environment in order to make it applicable to different business domains.

We are confident that we have met both these goals. In our model, a minimalist contract may be limited to header information including the contract participants and contractual-info describing foundational information. On the other hand, a complex unnoticed contractual relationship may be defined using our contract model, by exploiting the whole structure including contract-specific norms and institutional fact generating rules. The next steps of this work include exploring the developed contract model through different contracting scenarios.

Acknowledgments. This project is supported by FCT (Fundação para a Ciência e a Tecnologia) under Project POSC/EIA/57672/2004. Henrique Lopes Cardoso enjoys the FCT grant SFRH/BD/29773/2006.

References

[1] Arcos, J.L., Esteva, M., Noriega, P., Rodríguez-Aguilar, J.A., Sierra, C.: Environment engineering for multiagent systems. Engineering Applications of Artificial Intelligence 18, 191–204 (2005)

[2] Artikis, A., Pitt, J., Sergot, M.: Animated specifications of computational societies. In: Castelfranchi, C., Johnson, W.L. (eds.) International Joint Conference on Autonomous Agents and Multi-Agent Systems. Association for Computing Machinery, New York 10036-5701, United States, Bologna, Italy, pp. 1053–1062 (2002)

[3] Boella, G., van der Torre, L.: Contracts as Legal Institutions in Organizations of Autonomous Agents. In: Jennings, N.R., Sierra, C., Sonenberg, L., Tambe, M. (eds.) Third International Joint Conference on Autonomous Agents & Multi Agent Systems, pp. 948–955. ACM Press, New York (2004)

[4] Castelfranchi, C.: Engineering Social Order. In: Omicini, A., Tolksdorf, R., Zambonelli, F. (eds.) Engineering Societies in the Agents World, pp. 1–18. Springer, Berlin (2000)

[5] Craswell, R.: Contract Law: General Theories. In: Bouckaert, B., De Geest, G. (eds.) Encyclopedia of Law and Economics, pp. 1–24. Edward Elgar, Cheltenham (2000)

[6] Daskalopulu, A., Maibaum, T.: Towards Electronic Contract Performance. In: 12th International Conference and Workshop on Database and Expert Systems Applications, pp. 771–777. IEEE Computer Society Press, Los Alamitos (2001)

[7] Dignum, F., Broersen, J., Dignum, V., Meyer, J.-J.: Meeting the deadline: Why, when and how. In: Hinchey, M.G., Rash, J.L., Truszkowski, W.F., Rouff, C.A. (eds.) FAABS 2004. LNCS (LNAI), vol. 3228, pp. 30–40. Springer, Heidelberg (2004)

[8] Dignum, V., Meyer, J.-J.C., Dignum, F., Weigand, H.: Formal Specification of Interaction in Agent Societies. In: Hinchey, M., Rash, J., Truszkowski, W., Rouff, C., Gordon-Spears, D. (eds.) Formal Approaches to Agent-Based Systems, pp. 37–52. Springer, Heidelberg (2003)

[9] Fornara, N., Viganò, F., Colombetti, M.: Agent Communication and Institutional Reality. In: van Eijk, R.M., Huget, M.-P., Dignum, F. (eds.) Agent Communication: International Workshop on Agent Communication, pp. 1–17. Springer, Heidelberg (2005)

[10] Friedman-Hill, E.: Jess in Action, Manning Publications Co. (2003)

[11] García-Camino, A., Rodríguez-Aguilar, J.A., Sierra, C., Vasconcelos, W.: Norm-Oriented Programming of Electronic Institutions: A Rule-Based Approach. In: Noriega, P., Vázquez-Salceda, J., Boella, G., Boissier, O., Dignum, V., Fornara, N., Matson, E. (eds.) COIN 2006. LNCS (LNAI), vol. 4386, pp. 177–193. Springer, Heidelberg (2007)
[12] Jones, A., Sergot, M.: A Formal Characterisation of Institutionalised Power. Logic Journal of the IGPL 4, 427–443 (1996)
[13] Lopes Cardoso, H., Oliveira, E.: Electronic Institutions for B2B: Dynamic Normative Environments, Artificial Intelligence and Law (in press)
[14] 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, pp. 14–32. Springer, Heidelberg (2005)
[15] Searle, J.R.: The Construction of Social Reality. Free Press, New York (1995)
[16] 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, pp. 313–327. Springer, Heidelberg (2004)
[17] Weyns, D., Omicini, A., Odell, J.: Environment as a first class abstraction in multiagent systems. Journal of Autonomous Agents and Multi-Agent Systems 14, 5–30 (2007)