Monitoring Directed Obligations with Flexible Deadlines: A Rule-Based Approach

Henrique Lopes Cardoso and Eugénio Oliveira

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

Abstract. Real-world business relationships have an essentially cooperative nature. However, when modeling contractual norms using normative multi-agent systems, it is typical to give norms a strict and domain independent semantics. We argue that in B2B contract enactment cooperation should be taken into account when modeling contractual commitments through obligations. We introduce an approach to model such commitments based on directed obligations with time windows. Our proposal is based on authorizations granted at specific states of an obligation lifecycle model, made possible by handling deadlines in a flexible way. We formalize such obligations using linear temporal logic and provide an implementation to their semantics using a set of monitoring rules employed in a forward-chaining inference engine. We show, through experimentation, the correctness of the obtained monitoring tool in different contract enactment situations.

1 Introduction

Real-world business relationships have an essentially cooperative nature. When considering B2B Virtual Organizations, different enterprises share their own competences and skills in a regulated way, through commitments expressed as norms in contracts. The essence of such contracts is commitment [1]: contracts provide a legally binding agreement including legal sanctions in case of failure to honor commitments. However, the importance of successfully proceeding with business demands for flexibility of operations: contractors should try to facilitate the compliance of their partners. This common goal of conducting business is based on the fact that group success also benefits each partner's private goals. These goals are not limited to the ongoing business relationship, but may also concern future opportunities that may arise.

Multi-agent systems have been used to address B2B settings, where agents represent different enterprises and engage in (automated) contract negotiations. Furthermore, software frameworks are being designed and engineered that try to provide normative environments [2][3][4][5][6] enabling monitoring and enforcement of contractual norms. Nevertheless, many approaches to normative multi-agent systems are abstracted away from their potential application domain. As such, deontic operators used to describe norms are typically modeled with a

M. Baldoni et al. (Eds.): DALT 2009, LNAI 5948, pp. 51–67, 2010.

universal and domain independent semantics. For instance, deadline obligations are violated if the obliged action or state is not obtained before the deadline.

We argue that in some domains – such as in business contracts – such an approach is not desirable. For instance, the United Nations Convention on Contracts for the International Sale of Goods (CISG) [7] establishes what parties may do in case of deadline violations. In some cases they are allowed to fulfill their obligations after the deadline (Article 48), or even to extend deadlines with the allowance of their counterparties. Furthermore, a party may extend his counterparty's deadlines (Articles 47 and 63), which denotes a flexible and even cooperative facet of trade contracts.

In this paper we present and explore a different approach (in comparison with [8][9][10][11]) to the use of obligations in agent-based contracts. Following a cooperative business enactment principle, we argue that obligations should be directed (as in [11]), and that deadlines should be flexible (as they seem to be in the real world [7]). In section 2 we motivate our research with insights from a real-world legislation, and we present an approach to model contractual obligations with time windows. Our approach (more deeply analyzed in [12]) is based on authorizations, and includes a new lifecycle for directed obligations with temporal restrictions. A formalization is given using linear temporal logic. Section 3 describes a normative environment for norm monitoring, and includes a rule-based inference approach to our obligation semantics. Section 4 presents an implementation of the system, including monitoring rules, an example contract, and illustration of the system's response in different contract enactment situations. Finally, section 5 concludes and discusses our developments in the light of other approaches in the literature.

2 Modeling Contractual Obligations

Norms in MAS have been used for modeling regulated environments for agents. Deontic operators – obligation, permission and prohibition – form the basis for such approaches. In our case, we find obligations to be particularly relevant in the scope of business contracts.

Approaches to model obligations in MAS that have an implementation in mind typically consider two attributes: the bearer of the obligation and the deadline. We may represent such an obligation as $O_b(f, d)$: a *deadline obligation* indicating that agent b (the *bearer* of the obligation) is obliged to bring about fact f (a state of affairs) before deadline d (either a time reference or more generally defined as a state of affairs).

When recalling the usual approach to model the semantics of deadline obligations, as well as when presenting our proposal, we will make use of linear temporal logic (LTL) [13], with a discrete time model. Let $x = (s_0, s_1, s_2, ...)$ be a timeline, defined as a sequence of states s_i. The syntax $x \models p$ reads that p is true in timeline x. We write x^k to denote state s_k of x, and $x^k \models p$ to mean that p is true at state x^k. We use a weak version of the *before* LTL operator B, where q is not mandatory: $x \models (p\ B\ q)$ *iff* $\exists_j (x^j \models p \wedge \forall_{k<j} (x^k \models \neg q))$.

The semantics of deadline obligations has been studied before (e.g. [8][9]). The usual approach is to consider the following entailments:

– $O_b(f, d) \wedge (f\ B\ d) \models \mathit{Fulf}_b(f, d)$ — If the fact to bring about occurs before the deadline, the agent has *fulfilled* his obligation.
– $O_b(f, d) \wedge (d\ B\ f) \models \mathit{Viol}_b(f, d)$ — If the deadline occurs before the fact to bring about, the agent has *violated* his obligation.

The introduction of *Fulf* and *Viol* enables reasoning about the respective situations. The implementation of this semantics using forward-chaining rules has been studied in [9]. Although intuitive, this semantics is quite rigid in that violations are all defined in a universal way.

The analysis of contracts brings into discussion the notion of *directed obligations* [14]. An obligation $O_{b,c}(f)$ is seen as directed from agent b (the *bearer* responsible for fulfilling the obligation) to agent c (the *counterparty*). We interpret obligations of this kind as claims from counterparties to bearers (as in [11]): if b does not bring about f then c is *authorized* to react against b. Note that this reaction is discretionary, not mandatory.

In our approach we combine deadline obligations [8] with directed obligations [14][11], in order to obtain a more precise definition of when it is that a counterparty may claim against the inability of a bearer to fulfill the obligation. We will motivate and formalize the notion of *directed deadline obligation* – $O_{b,c}(f, d)$: agent b is obliged towards agent c to bring about f before d. An extension of directed (contractual) obligations with temporal restrictions is also introduced in [10], but that approach is based on a rigid model of violations, in that they are automatically obtained at the deadline. In our approach deadlines have a distinct role in the semantics of obligations. We will introduce the notion of *deadline violation* (as opposed to obligation violation) in order to obtain a flexible approach to handle non-ideal situations: each deadline violation is different, as each may have a different impact on the ongoing business, and each occurs between a specific pair of agents with a unique trust relationship.

2.1 Directed Obligations with Time Windows

When specifying norms in contracts, deadline handling is central to define the semantics of contractual obligations. In order to motivate our approach, we take some inspiration from the United Nations Convention on Contracts for the International Sale of Goods (CISG) [7]. Some excerpts of this legislation are included in the following discussion.

> *Article 48: (1) [...] the seller may, even after the date for delivery, remedy at his own expense any failure to perform his obligations, if he can do so without unreasonable delay [...]; (2) If the seller requests the buyer to make known whether he will accept performance and the buyer does not comply with the request within a reasonable time, the seller may perform within the time indicated in his request. [...]*

This means that even though a deadline has been violated, the bearer may still be entitled to fulfill *the same* obligation. This kind of delay is also called a *grace period*: a period beyond a due date during which an obligation may be met without penalty or cancellation.

In fact, the successful enactment of a contract is dependent on the need to make contractual provisions performable in a flexible way:

> *Article 47: (1) The buyer may fix an additional period of time of reasonable length for performance by the seller of his obligations.*

> *Article 63: (1) The seller may fix an additional period of time of reasonable length for performance by the buyer of his obligations.*

These articles emphasize the need for flexible deadlines. Note that the counterparty's benevolence on conceding an extended deadline to the bearer does not prescribe a new obligation; instead, *the same* obligation may be fulfilled within a larger time window. Furthermore, it is also in the counterparty's best interest that this option is available, given the importance of reaching success in the performance of the contract.

In some other cases, a party may decide that the non-fulfillment of an obligation should be handled in a more strict way. The CISG convention specifies conditions for cancelling a contract in case of breach:

> *Article 49: (1) The buyer may declare the contract avoided: (a) if the failure by the seller to perform any of his obligations [...] amounts to a fundamental breach of contract; [...]; (2) However, in cases where the seller has delivered the goods, the buyer loses the right to declare the contract avoided unless he does so: (a) in respect of late delivery, within a reasonable time after he has become aware that delivery has been made; [...]*

> *Article 64: (1) The seller may declare the contract avoided: (a) if the failure by the buyer to perform any of his obligations [...] amounts to a fundamental breach of contract; [...]; (2) However, in cases where the buyer has paid the price, the seller loses the right to declare the contract avoided unless he does so: (a) in respect of late performance by the buyer, before the seller has become aware that performance has been rendered; [...]*

These articles allow contract termination in both non-performance and late performance cases. However, the second case is limited to the awareness of the offended party.

The deadline approach is often taken to be appropriate for specifying temporal restrictions on obligations. However, in certain cases a time window should be provided. In international trade transactions, for instance, storage costs may be relevant. Also, perishable goods should be delivered only when they are needed, not before.

> *Article 52: (1) If the seller delivers the goods before the date fixed, the buyer may take delivery or refuse to take delivery.*

Therefore, anticipated fulfillments are not always welcome. We find it necessary to include a variation of directed deadline obligations, to which we add a *liveline*: a time reference after which the obligation should be fulfilled. In this case we have $O_{b,c}(f, l, d)$: agent b is obliged towards agent c to bring about f between l (a liveline) and d (a deadline).

The intuitive semantics of directed deadline obligations and directed obligations with liveline and deadline are illustrated in Figures 1 and 2. The shaded areas represent the period of time within which the achievement of f will certainly bring a fulfillment of the obligation. The region to the left of l (Figure 2) entitles c to react if f is accomplished; also, regions to the right of d in both figures indicate that counterparty c is entitled to react if f is not accomplished. However, as long as no reaction is taken, b can still fulfill his obligation.

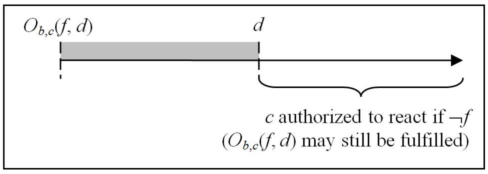

Fig. 1. Directed obligation with deadline

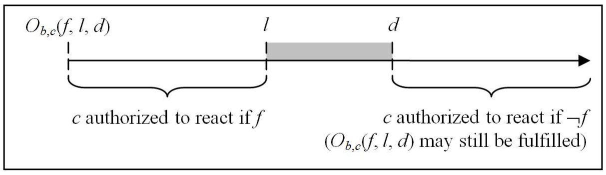

Fig. 2. Directed obligation with liveline and deadline

2.2 Formalization with LTL

Following the discussion above, in Table 1 we identify the possible states for an obligation, together with the elements we shall use to signal some of those states (when obtained, these elements are supposed to persist over time).

We now proceed to formalizing each type of obligation using LTL.

Directed Deadline Obligations. Figure 3 illustrates, by means of a state transition diagram, the lifecycle of directed deadline obligations. We take obligations as being prescribed from conditional norms; the confirmation of the norm's conditions will change the prescribed obligation's state from inactive to active. The obligation is also automatically pending, since it may be legitimately fulfilled right away. We set the obligation to have a violated deadline – $DViol_{b,c}(f, d)$ – when the deadline occurs before the obliged fact. The counterparty's reaction

Table 1. Obligation states

inactive:	the obligation is not yet in effect, but will eventually be prescribed by a norm
active:	the obligation was prescribed by a norm: $O_{b,c}(f, d)$ or $O_{b,c}(f, l, d)$
pending:	the obligation may be fulfilled from now on
liveline violation:	the fact being obliged has been brought ahead of time: $LViol_{b,c}(f, l, d)$
deadline violation:	the fact being obliged should have been brought already: $DViol_{b,c}(f, d)$ or $DViol_{b,c}(f, l, d)$
fulfilled:	the obligation was fulfilled: $Fulf_{b,c}(f, d)$ or $Fulf_{b,c}(f, l, d)$
violated:	the obligation was violated and cannot be fulfilled anymore: $Viol_{b,c}(f, d)$ or $Viol_{b,c}(f, l, d)$

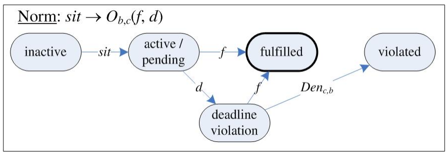

Fig. 3. Lifecycle of a directed deadline obligation

to a deadline violation will only change the obligation's state if the option is to deem the obligation as violated, by *denouncing* this situation. For this we introduce the element $Den_{c,b}(f, d)$, which is a denounce from agent c towards agent b regarding the failure of the latter to comply with his obligation to bring about f before d.

The lifecycle of directed deadline obligations is formalized as follows:

- $O_{b,c}(f, d) \wedge (f\ B\ d) \models Fulf_{b,c}(f, d)$
- $O_{b,c}(f, d) \wedge (d\ B\ f) \models DViol_{b,c}(f, d)$
- $DViol_{b,c}(f, d) \wedge (f\ B\ Den_{c,b}(f, d)) \models Fulf_{b,c}(f, d)$
- $DViol_{b,c}(f, d) \wedge (Den_{c,b}(f, d)\ B\ f) \models Viol_{b,c}(f, d)$

Directed Obligations with Liveline and Deadline. Figure 4 contains the state transition diagram for directed obligations with liveline and deadline. In this case, the obligation will only be pending when l arises, since only then it may be fulfilled in a way that is compliant with the terms of the contract. We have now two kinds of temporal violations: liveline violations of the form $LViol_{b,c}(f, l, d)$ and deadline violations of the form $DViol_{b,c}(f, l, d)$. In both cases, a denounce ($Den_{c,b}(f, l, d)$) may establish the obligation as violated, if issued before l or f, respectively.

The lifecycle of directed obligations with liveline and deadline is formalized as follows:

- $O_{b,c}(f, l, d) \wedge (f\ B\ l) \models LViol_{b,c}(f, l, d)$
- $LViol_{b,c}(f, l, d) \wedge (l\ B\ Den_{c,b}(f, l, d)) \models Fulf_{b,c}(f, l, d)$

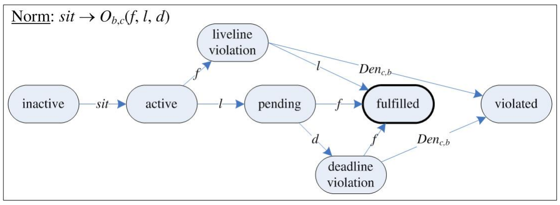

Fig. 4. Lifecycle of a directed obligation with liveline and deadline

- $LViol_{b,c}(f, l, d) \wedge (Den_{c,b}(f, l, d)\ B\ l) \models Viol_{b,c}(f, l, d)$
- $O_{b,c}(f, l, d) \wedge (l\ B\ f) \wedge (f\ B\ d) \models Fulf_{b,c}(f, l, d)$
- $O_{b,c}(f, l, d) \wedge (d\ B\ f) \models DViol_{b,c}(f, l, d)$
- $DViol_{b,c}(f, l, d) \wedge (f\ B\ Den_{c,b}(f, l, d)) \models Fulf_{b,c}(f, l, d)$
- $DViol_{b,c}(f, l, d) \wedge (Den_{c,b}(f, l, d)\ B\ f) \models Viol_{b,c}(f, l, d)$

3 A Normative Environment for Monitoring Norms

In this section we describe our efforts to bring the formalizations presented above towards an implementation of a normative environment with norm monitoring in place. The normative environment is composed of three main ingredients: *normative state*, *institutional rules* and *norms*. The normative state is a set of fully-grounded facts describing relevant events. Institutional rules are rules that manipulate the normative state. Norms are a special kind of rules, in that they are used to prescribe behavior. The next subsections detail each of these elements.

3.1 Normative State

In business contracts it is common to have deadlines that are dependent on the fulfillment date of other obligations. Therefore, instead of having fixed (absolute) dates, these may at times be relative, calculated according to other events. CISG [7] expresses this by saying that dates can be *determinable* from the contract:

> *Article 33: The seller must deliver the goods: (a) if a date is fixed by or determinable from the contract, on that date; (b) if a period of time is fixed by or determinable from the contract, at any time within that period [...]*

> *Article 59: The buyer must pay the price on the date fixed by or determinable from the contract [...]*

It is therefore useful to timestamp each event. For this reason, we will use explicit time references in the constituting elements of the normative state. Furthermore, rules and norms will make use of these time references.

We will call each element of the normative state an *institutional reality element* (*IRE*). These elements allow us to record all relevant events regarding the social context that is being monitored. We distinguish several kinds of *IRE*, as shown in Table 2.

Table 2. Institutional reality elements

$\mathit{Ifact}(f)^t$:	fact f is institutionally recognized as being the case at time t
$O_{b,c}(f,l,d)^t$:	agent b is obliged, since t, towards agent c to bring about f between l and d
$\mathit{LViol}_{b,c}(f,l,d)^t$:	there was a liveline violation at time t (fact f has been brought too early)
$\mathit{DViol}_{b,c}(f,l,d)^t$:	there was a deadline violation at time t (fact f should have been brought already)
$\mathit{Fulf}_{b,c}(f,l,d)^t$:	agent b has fulfilled, at time t, his obligation to bring about f between l and d
$\mathit{Viol}_{b,c}(f,l,d)^t$:	agent b has violated, at time t, his obligation to bring about f between l and d
$\mathit{Den}_{c,b}(f,l,d)^t$:	agent c has denounced, at time t, the failure of agent b to bring about f between l and d
$\mathit{Time}(t)$:	instant t has elapsed

An institutional fact (*Ifact*) is a piece of evidence that certifies the occurrence of an event, where such event (a consequence of an agent action) may be external to the normative environment itself, while being part of the social context it is supposed to regulate. Time elements are used to signal the reach of contractually relevant time instants (namely those concerning livelines and deadlines).

3.2 Rules and Norms

Some of the IRE's that are part of the normative state are interrelated, in the sense that some IRE's are obtained from other IRE's. These interrelations are captured with institutional rules and norms. In the literature [15][16] a distinction has been made between regulative and constitutive rules. We see norms as regulative rules, that is, rules that change the normative positions of agents, e.g. by prescribing obligations. Our institutional rules allow us to iterate through institutional facts and to further obtain IRE's other than obligations. For instance, institutional rules may be defined to indicate how a denouncement may be obtained from an institutional fact. This kind of inference has therefore a constitutive nature. In this paper we will concentrate on a particular kind of institutional rules: monitoring rules, that are used to implement the semantics of directed obligations with liveline and deadline.

Both institutional rules and norms have a rule-based materialization. Their left-hand-sides (conditions) are composed of (possibly negated) conjunctions of patterns of IRE's, which may contain (universally quantified) variables; restrictions may be imposed on such variables through relational conditions. The right-hand-sides (conclusions) of institutional rules are conjunctions of non-deontic IRE's which are allowed to contain bounded variables; the right-hand-sides of norms are conjunctions of deontic IRE's (obligations) which are allowed to contain bounded variables.[1]

[1] A detailed formalization can be found in [9], with a more complex model that includes context-dependent information elements, which is out of the scope of this paper.

When the conditions of a rule (or norm) match the normative state using a first-order logic substitution Θ, and if all the relational conditions over variables hold, the atomic formulae obtained by applying Θ to the consequent of the rule are added to the normative state as fully-grounded elements.

3.3 Monitoring Rules

The LTL logical relationships in Section 2.2 provided us a formalism to define directed obligations with liveline and deadline. However, in order to monitor contracts at run-time, we need to ground this semantics into a reasoning engine capable of responding to events in a timely fashion. Elements describing obligation states should allow us to reason about those states *as soon as they occur*. A natural choice we have made before [9] is the use of a rule-based inference engine, with which the following (forward-chaining) rules can be defined to implement the semantics of directed obligations with liveline and deadline:

- $O_{b,c}(f,l,d)^i \wedge Ifact(f)^t \wedge t < l \rightarrow LViol_{b,c}(f,l,d)^t$
- $LViol_{b,c}(f,l,d)^i \wedge Time(l) \wedge \neg(Den_{c,b}(f,l,d)^u \wedge u < l) \rightarrow Fulf_{b,c}(f,l,d)^l$
- $LViol_{b,c}(f,l,d)^i \wedge Den_{c,b}(f,l,d)^u \wedge u < l \rightarrow Viol_{b,c}(f,l,d)^u$
- $O_{b,c}(f,l,d)^i \wedge Ifact(f)^t \wedge l < t \wedge t < d \rightarrow Fulf_{b,c}(f,l,d)^t$
- $O_{b,c}(f,l,d)^i \wedge Time(d) \wedge \neg(Ifact(f)^t \wedge t < d) \rightarrow DViol_{b,c}(f,l,d)^d$
- $DViol_{b,c}(f,l,d)^i \wedge Ifact(f)^t \wedge \neg(Den_{c,b}(f,l,d)^u \wedge u < t) \rightarrow Fulf_{b,c}(f,l,d)^t$
- $DViol_{b,c}(f,l,d)^i \wedge Den_{c,b}(f,l,d)^u \wedge \neg(Ifact(f)^t \wedge t < u) \rightarrow Viol_{b,c}(f,l,d)^u$

These rules take into account the time instants when each institutional reality element occurs in order to assert other elements with accurate timestamps. Notice the use of relational conditions in order to assess the temporal ordering of events that may match each rule's conditions.

In the next section we provide an implementation for these rules using a forward-chaining rule-based inference engine.

4 Implementation with Jess

We have chosen Jess[2] [17] to implement our norm monitoring system. Jess is a very efficient rule engine based on the Rete algorithm for pattern matching. We start by defining appropriate templates (through `deftemplate` constructs) for each type of element in the normative state. Jess facts follow a frame-like notation, in which each fact has associated slots to be filled in. Template inheritance is possible via the `extends` keyword. We have:

```
(deftemplate institutional-reality-element
  (slot when) )

(deftemplate ifact extends institutional-reality-element
  (multislot fact) )
```

[2] The code presented in this section includes some simplifications in order to make it more simple to understand.

```
(deftemplate obligation extends institutional-reality-element
  (slot bearer) (slot counterparty) (multislot fact)
  (slot liveline) (slot deadline) )

(deftemplate liveline-violation extends institutional-reality-element
  (slot obl) )

(deftemplate deadline-violation extends institutional-reality-element
  (slot obl) )

(deftemplate fulfillment extends institutional-reality-element
  (slot obl) )

(deftemplate violation extends institutional-reality-element
  (slot obl) )

(deftemplate denounce extends institutional-reality-element
  (slot obl) )

(deftemplate time extends institutional-reality-element)

(deftemplate cancel-contract extends institutional-reality-element)
```

For simplification, we included an obligation reference inside some templates. The `time` template is used to assert the occurrence of time events (associated with livelines and deadlines), which is done by scheduling alerts using a system clock. The `cancel-contract` template enables contract cancellation assertions, and will be used in our contract example below.

4.1 Monitoring Rules

Implementing monitoring rules in Jess is straightforward. A Jess rule is written in the form *LHS* => *RHS*, where *LHS* includes fact patterns that will be matched against facts in working memory (our normative state). The *RHS* indicates actions to execute (such as asserting new facts) when the rule is fired. The following rules (defined with `defrule` constructs) translate directly from the monitoring rules shown in Section 3.3 (identifiers starting with a question mark are variables).

```
(defrule detect-liveline-violation
  ?obl <- (obligation (fact $?f) (liveline ?l))
  ?ifa <- (ifact (fact $?f) {when < ?l})
  =>
  (assert (liveline-violation (obl ?obl) (when ?ifa.when))) )

(defrule detect-early-fulfillment
  ?lviol <- (liveline-violation (obl ?obl))
  ?obl <- (obligation (liveline ?l))
  (time (when ?l))
  (not (denounce (obl ?obl) {when < ?l}))
  =>
  (assert (fulfillment (obl ?obl) (when ?l))) )
```

```
(defrule detect-violation-before-liveline
  ?lviol <- (liveline-violation (obl ?obl))
  ?obl <- (obligation (liveline ?l))
  ?den <- (denounce (obl ?obl) {when < ?l})
  =>
  (assert (violation (obl ?obl) (when ?den.when))) )

(defrule detect-fulfillment
  ?obl <- (obligation (fact $?f) (liveline ?l) (deadline ?d))
  ?ifa <- (ifact (fact $?f) {when >= ?l && when <= ?d))
  =>
  (assert (fulfillment (obl ?obl) (when ?ifa.when))) )

(defrule detect-deadline-violation
  ?obl <- (obligation (fact $?f) (deadline ?d))
  (time (when ?d))
  (not (ifact (fact $?f) {when <= ?d}))
  =>
  (assert (deadline-violation (obl ?obl) (when ?d))) )

(defrule detect-belated-fulfillment
  (deadline-violation (obl ?obl))
  ?obl <- (obligation (fact $?f))
  (ifact (fact $?f) (when ?t))
  (not (denounce (obl ?obl) {when <= ?t}))
  =>
  (assert (fulfillment (obl ?obl) (when ?t))) )

(defrule detect-violation-after-deadline
  (deadline-violation (obl ?obl))
  ?obl <- (obligation (fact $?f))
  (denounce (obl ?obl) (when ?u))
  (not (ifact (fact $?f) {when < ?u}))
  =>
  (assert (violation (obl ?obl) (when ?u))) )
```

In order to determine denouncements from institutional facts, we define the following (constitutive) institutional rule:

```
(defrule denounce-recognition
    (ifact (fact denounce $?f) (when ?w))
    ?obl <- (obligation (fact $?f))
    =>
    (assert (denounce (obl ?obl) (when ?w))) )
```

These rules enable us to monitor the compliance of agents with their obligations. Norms are used to prescribe such obligations, by asserting them into the normative state. The normative environment's monitoring capabilities may be used as a tool for alerting agents when certain contract-related events occur. Further rules may be defined with such a purpose. The *RHS* of Jess rules may include function calls that implement the desired level of responsiveness of the normative environment in which notifications are concerned.

4.2 Example Contract

In this section we show a simple example where the concept of flexible deadlines is exploited in an electronically supervised business relationship. We have a contract between two agents, say `B` and `S`, wherein `S` commits to supply, whenever ordered, good `X` for 7.5 per unit.

The norms below (implemented as Jess rules) define the contractual relationship and are included in the institutional normative environment for monitoring purposes. Agent `S` is supposed to deliver the ordered goods between 3 to 5 days after the order (norm `n1`), and agent `B` shall pay within 30 days (norm `n2`). Furthermore, if agent `B` does not pay in due time, he will incur in a penalty consisting of an obligation to pay an extra 10% on the order total (norm `n3`). Finally, if agent `S` violates his obligation to deliver, the contract shall be canceled (norm `n4`).

```
(defrule n1
  (ifact (fact order item X quantity ?q) (when ?w))
  =>
  (assert
   (obligation (bearer S) (counterparty B) (fact delivery X qt ?q)
               (liveline (+ ?w 3)) (deadline (+ ?w 5)) (when ?w)) ) )

(defrule n2
  (fulfillment (obl ?obl) (when ?w))
  ?obl <- (obligation (fact delivery X qt ?q))
  =>
  (assert
   (obligation (bearer B) (counterparty S) (fact payment (* ?q 7.5))
               (liveline ?w) (deadline (+ ?w 30)) (when ?w)) ) )

(defrule n3
  (deadline-violation (obl ?obl))
  ?obl <- (obligation (fact payment ?p) (deadline ?d))
  =>
  (assert
   (obligation (bearer B) (counterparty S) (fact payment (* ?p 0.10))
               (liveline ?d) (deadline (+ ?d 30)) (when ?d)) ) )

(defrule n4
  (violation (obl ?obl) (when ?w))
  ?obl <- (obligation (fact delivery X qt ?q))
  =>
  (assert (cancel-contract (when ?w)) ) )
```

In this example we can see that interests applied on payments are automatic once deadline violations are detected (norm `n3`). On the other hand, a contract cancellation (norm `n4`) requires that agent `B` denounces the inability of agent `S` to

Enactment 1: everything goes as agreed

```
f-6  * (ifact (when 1) (fact order item X quantity 5))
f-7    (obligation (when 1) (bearer S) (counterparty B)
                   (fact delivery X qt 5) (liveline 4) (deadline 6))
f-11   (time (when 4))
f-13 * (ifact (when 5) (fact delivery X qt 5))
f-14   (fulfillment (when 5) (obl <Fact-7>))
f-15   (obligation (when 5) (bearer B) (counterparty S)
                   (fact payment 37.5) (liveline 5) (deadline 35))
f-16   (time (when 5))
f-18   (time (when 6))
f-26 * (ifact (when 13) (fact payment 37.5))
f-27   (fulfillment (when 13) (obl <Fact-15>))
f-50   (time (when 35))
```

Fig. 5. A perfect enactment

fulfill the delivery. It is therefore up to agent B whether to wait further and accept a delayed delivery or not. If the agreed upon contract conditions are important enough, allowing a counterparty deviation (and hence taking a cooperative attitude regarding the compliance of the contract) may be a good decision.

4.3 Contract Enactments

In this section we show the outcomes of applying monitoring rules and contractual norms in different contract enactment situations, taking into account the contract presented above.

Figures 5, 6 and 7 show the response of monitoring rules to different enactment situations. The listings in these figures show the normative state after contract enactment, including relevant *IRE* that are produced by rules and norms, together with institutional facts originated by agent actions (these are marked with an asterisk). Time events (associated with livelines and deadlines) triggered by a system clock are also shown.

Figure 5 shows the normative state after a perfect contract enactment, where everything goes as agreed. No temporal violations are detected in this case, since agents abide to their obligations. Figure 6 depicts several enactment outcomes in which delivery problems are detected. In enactments 2 and 3 the delivery liveline or deadline is violated (and detected by rules adding `f-10` and `f-15`, respectively), while the counterparty does not denounce this situation. Enactment 4 shows a situation in which the counterparty chooses to denounce (`f-17`) a deadline violation (detected in `f-14`); as indicated in contractual norm `n4`, this results in a contract cancellation (`f-20`). Finally, figure 7 shows an enactment in which the payment deadline was violated, bringing an interest rate to be applied according to contractual norm `n3`. Agent `B` eventually payed both the price (`f-57`) and the interest rate (`f-58`).

Enactment 2: delivery liveline violation without denounce

```
f-6  * (ifact (when 1) (fact order item X quantity 5))
f-7    (obligation (when 1) (bearer S) (counterparty B)
                   (fact delivery X qt 5) (liveline 4) (deadline 6))
f-9  * (ifact (when 2) (fact delivery X qt 5))
f-10   (liveline-violation (when 2) (obl <Fact-7>))
f-13   (time (when 4))
f-14   (fulfillment (when 4) (obl <Fact-7>))
f-15   (obligation (when 4) (bearer B) (counterparty S)
                   (fact payment 37.5) (liveline 4) (deadline 34))
f-18   (time (when 6))
f-26 * (ifact (when 13) (fact payment 37.5))
f-27   (fulfillment (when 13) (obl <Fact-15>))
f-49   (time (when 34))
```

Enactment 3: delivery deadline violation without denounce

```
f-6  * (ifact (when 1) (fact order item X quantity 5))
f-7    (obligation (when 1) (bearer S) (counterparty B)
                   (fact delivery X qt 5) (liveline 4) (deadline 6))
f-11   (time (when 4))
f-14   (time (when 6))
f-15   (deadline-violation (when 6) (obl <Fact-7>))
f-19 * (ifact (when 9) (fact delivery X qt 5))
f-20   (fulfillment (when 9) (obl <Fact-7>))
f-21   (obligation (when 9) (bearer B) (counterparty S)
                   (fact payment 37.5) (liveline 9) (deadline 39))
f-22   (time (when 9))
f-27 * (ifact (when 13) (fact payment 37.5))
f-28   (fulfillment (when 13) (obl <Fact-21>))
f-55   (time (when 39))
```

Enactment 4: delivery deadline violation with denounce

```
f-5  * (ifact (when 1) (fact order item X quantity 5))
f-6    (obligation (when 1) (bearer S) (counterparty B)
                   (fact delivery X qt 5) (liveline 4) (deadline 6))
f-10   (time (when 4))
f-13   (time (when 6))
f-14   (deadline-violation (when 6) (obl <Fact-6>))
f-17 * (ifact (when 8) (fact denounce delivery X qt 5))
f-18   (denounce (when 8) (obl <Fact-6>))
f-19   (violation (when 8) (obl <Fact-6>))
f-20   (cancel-contract (when 8))
```

Fig. 6. Delivery problems

<u>Enactment 5</u>: payment deadline violation

```
f-7  * (ifact (when 1) (fact order item X quantity 5))
f-8    (obligation (when 1) (bearer S) (counterparty B)
                   (fact delivery X qt 5) (liveline 4) (deadline 6))
f-12   (time (when 4))
f-14 * (ifact (when 5) (fact delivery X qt 5))
f-15   (fulfillment (when 5) (obl <Fact-8>))
f-16   (obligation (when 5) (bearer B) (counterparty S)
                   (fact payment 37.5) (liveline 5) (deadline 35))
f-17   (time (when 5))
f-19   (time (when 6))
f-49   (time (when 35))
f-50   (deadline-violation (when 35) (obl <Fact-16>))
f-51   (obligation (when 35) (bearer B) (counterparty S)
                   (fact payment 3.75) (liveline 35) (deadline 65))
f-57 * (ifact (when 40) (fact payment 3.75))
f-58 * (ifact (when 40) (fact payment 37.5))
f-59   (fulfillment (when 40) (obl <Fact-16>))
f-60   (fulfillment (when 40) (obl <Fact-51>))
f-86   (time (when 65))
```

Fig. 7. Payment problems

5 Conclusions

In B2B relationships contracts specify, through obligations, the interdependencies between different partners, and provide legal options to which parties can resort in case of conflict. However, when this joint activity aims at pursuing a common goal, the successful performance of business benefits all involved parties. Therefore, when developing automated monitoring tools, one should take into account that partners may be cooperative enough to allow counterparty deviations. Taking this into account, we have developed a novel model for contractual obligations, where these are seen as either *directed deadline obligations* or *directed obligations with liveline and deadline*. The directed aspect concerns the need to identify the agent who will be authorized to react in case of non-fulfillment. We link authorizations with a flexible model of livelines and deadlines. Obligation violations are dependent on the counterparty motivation to claim them. Using flexible deadlines ensures a degree of freedom for agents to make decisions in the execution phase of contracts, which is important for dealing with business uncertainty. Our approach is based on real-world evidence from business contracts (namely the United Nations Convention on Contracts for the International Sale of Goods [7]), which denotes a flexible and even cooperative facet of trade contracts.

Most implementations of norms in multi-agent systems ignore the need for having directed obligations from bearers to counterparties. The most likely reason for this is that in those approaches obligations are seen as (implicitly) directed from an agent to the normative system itself. It is up to the system (e.g.

an electronic organization [18] or an electronic institution [19]) to detect violations and to enforce the norms which are designed into the environment (in some cases they are even regimented in such a way that violation is not possible). On the contrary, our flexible approach towards an Electronic Institution [5][20] allows agents to define the norms that will regulate their mutual commitments.

An issue that we have not included in our model is the need for agents to communicate their intentions regarding an obligation with a violated deadline. In fact, CISG's Article 48 seems to go in this direction, in order to protect the bearer's efforts toward a late fulfillment of the obligation. This concern has been taken into account in [21], where a contract fulfillment protocol demanding agents to communicate their intentions drives an obligation lifecycle model. The states of this lifecycle are obtained according to the performance of a contractual relationship.

Our implementation using a forward-chaining rule-based approach is applicable to run-time monitoring of contracts. A requirement of this kind of usage however is that events are reported in a timely fashion to the normative environment. We assume that agents are interested in publicizing their abidance to commitments. The monitoring capabilities of our implementation may, however, be used as a tool to alert agents when certain contract-related events are eminent, such as a forthcoming deadline. Jess [17] allows for an easy integration of our monitoring rules implementation with other rules including function calls that address the level of responsiveness that is intended.

We have shown how the normative environment may effectively monitor contract enactment at run-time. Monitoring rules may also be used *a posteriori*, in order to check off-line if contractual norms were indeed followed by every partner. In this case, after collecting all events concerning a contract, the inference engine may run in order to check if the contract was enacted in a conforming way.

Acknowledgments. The first author is supported by FCT (Fundação para a Ciência e a Tecnologia) under grant SFRH/BD/29773/2006.

References

1. Masten, S.E.: Contractual choice. In: Bouckaert, B., De Geest, G. (eds.) Encyclopedia of Law and Economics. Volume Volume III: The Regulation of Contracts, pp. 25–45. Edward Elgar, Cheltenham (2000)
2. Arcos, J.L., Esteva, M., Noriega, P., Rodríguez-Aguilar, J.A., Sierra, C.: Environment engineering for multiagent systems. Engineering Applications of Artificial Intelligence 18(2), 191–204 (2005)
3. Artikis, A., Sergot, M., Pitt, J.: Specifying norm-governed computational societies. ACM Transactions on Computational Logic 10(1), 1:1–1:42 (2009)
4. Fornara, N., Colombetti, M.: Specifying and enforcing norms in artificial institutions. In: Padgham, P., Müller, P. (eds.) 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008), Estoril, Portugal, pp. 1481–1484 (2008)
5. Lopes Cardoso, H., Oliveira, E.: Electronic institutions for b2b: Dynamic normative environments. Artificial Intelligence and Law 16(1), 107–128 (2008)

6. Modgil, S., Faci, N., Meneguzzi, F., Oren, N., Miles, S., Luck, M.: A framework for monitoring agent-based normative systems. In: Decker, S., Sierra, C. (eds.) Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems. IFAAMAS, Budapest, Hungary, pp. 153–160 (2009)
7. UNCITRAL: United nations convention on contracts for the international sale of goods, cisg (1980)
8. Broersen, J., Dignum, F., Dignum, V., Meyer, J.J.C.: Designing a deontic logic of deadlines. In: Lomuscio, A., Nute, D. (eds.) DEON 2004. LNCS (LNAI), vol. 3065, pp. 43–56. Springer, Heidelberg (2004)
9. Lopes Cardoso, H., Oliveira, E.: A context-based institutional normative environment. In: Hubner, J., Matson, E., Boissier, O., Dignum, V. (eds.) Coordination, Organizations, Institutions, and Norms in Agent Systems IV. LNCS (LNAI), vol. 5428, pp. 140–155. Springer, Heidelberg (2009)
10. Ryu, Y.U.: Relativized deontic modalities for contractual obligations in formal business communication. In: 30th Hawaii International Conference on System Sciences (HICSS), Hawaii, USA, vol. 4, pp. 485–493 (1997)
11. Tan, Y.H., Thoen, W.: Modeling directed obligations and permissions in trade contracts. In: Proceedings of the Thirty-First Annual Hawaii International Conference on System Sciences, vol. 5. IEEE Computer Society, Los Alamitos (1998)
12. Lopes Cardoso, H., Oliveira, E.: Directed deadline obligations in agent-based business contracts. In: Artikis, A., Vasconcelos, W. (eds.) AAMAS 2009 Workshop on Coordination, Organization, Institutions and Norms in agent systems (COIN), Budapest, Hungary, pp. 77–92 (2009)
13. Emerson, E.A.: Temporal and modal logic. In: Leeuwen, J.v. (ed.) Handbook of Theoretical Computer Science. Formal Models and Sematics, vol. B, pp. 995–1072. North-Holland Pub. Co./MIT Press (1990)
14. Herrestad, H., Krogh, C.: Obligations directed from bearers to counterparties. In: Proceedings of the 5th international conference on Artificial intelligence and law, College Park, Maryland, United States, pp. 210–218. ACM, New York (1995)
15. Searle, J.R.: The Construction of Social Reality. Free Press, New York (1995)
16. Boella, G., van der Torre, L.: Regulative and constitutive norms in normative multiagent systems. In: Dubois, D., Welty, C., Williams, M.A. (eds.) Ninth International Conference on Principles of Knowledge Representation and Reasoning, pp. 255–266. AAAI Press, Whistler (2004)
17. Friedman-Hill, E.: Jess in Action. Manning Publications Co. (2003)
18. Vázquez-Salceda, J., Dignum, F.: Modelling electronic organizations. In: Mařík, V., Müller, J.P., Pěchouček, M. (eds.) CEEMAS 2003. LNCS (LNAI), vol. 2691, pp. 584–593. Springer, Heidelberg (2003)
19. Esteva, M., Rodríguez-Aguilar, J.A., Sierra, C., Garcia, P., Arcos, J.L.: On the formal specifications of electronic institutions. In: Dignum, F., Sierra, C. (eds.) Agent-mediated Electronic commerce: The European AgentLink Perspective. LNCS (LNAI), vol. 1991, pp. 126–147. Springer, Heidelberg (2001)
20. Lopes Cardoso, H., Oliveira, E.: Norm defeasibility in an institutional normative framework. In: Proceedings of The 18th European Conference on Artificial Intelligence (ECAI 2008), Patras, Greece, pp. 468–472. IOS Press, Amsterdam (2008)
21. Sallé, M.: Electronic contract framework for contractual agents. In: Cohen, R., Spencer, B. (eds.) Canadian AI 2002. LNCS (LNAI), vol. 2338, pp. 349–353. Springer, Heidelberg (2002)