A Real-Time Semantics for Norms with Deadlines

Koen V. Hindriks
Delft University of Technology
Delft, The Netherlands
k.v.hindriks@tudelft.nl

M. Birna van Riemsdijk
Delft University of Technology
Delft, The Netherlands
m.b.vanriemsdijk@tudelft.nl

ABSTRACT

Norms have been proposed as a way to regulate multi-agent systems. In order to operationalize norms, several computational frameworks have been proposed for programming norm-governed agent organizations. It has been argued that in such systems it is essential that norms, in particular those giving rise to achievement obligations, have deadlines. In this paper we propose a novel semantic framework that takes into account and formalizes *real-time* aspects of such norms with deadlines. The framework introduced provides a semantics for norms with real-time deadlines that is a conservative extension of more traditional transition systems semantics that has been used for specifying multi-agent systems. Our framework thus provides a natural extension for formalizing multi-agent systems with norms that have real-time deadlines. We address several important aspects of semantics of norms with deadlines such as deadline termination and, in particular, investigate the issue of deadline shifting that arises naturally in a real-time setting as a result of interactions between norms. A new normative model is presented for handling such interactions. We present several formal results showing that our semantics corresponds with basic intuitions that any operational semantics for norms with (real-time) deadlines should satisfy, and that it is well-defined.

Categories and Subject Descriptors

I.2.5 [**Artificial Intelligence**]: Programming Languages and Software

Keywords

Real-Time Semantics; Norms; Deadlines

1. INTRODUCTION

A system of autonomous agents may exhibit undesirable or ineffective behavior if no form of regulation is imposed. An important line of research that addresses this issue is work on *normative systems*, in which norms govern the behavior of a multi-agent system (MAS). Norms describe how agents should ideally behave. In this paper we are concerned with *achievement obligations* (where an agent is obligated to achieve something in the future that is not already true now).

It has been argued in research on deontic logic and elsewhere that achievement obligations need a *deadline* [7, 3, 15]. For example, a norm like "if a buyer pays for a product, the seller is obligated to hand out a receipt" without a deadline is too weak. The norm does not specify when the seller should hand out the receipt, and so the seller could take an indefinite amount of time to do this. Put differently, the seller could never be viewed as violating this norm, because she can always say she will do this sometime in the future. Deadlines thus are an essential aspect of achievement obligations to motivate an agent to meet its obligations and such obligations should therefore be endowed with deadlines in order to effectively regulate agent behavior.

We focus on quantitative or *real-time deadlines* which refer to time (e.g., a customer is obligated to pay within 10 days of issuing the invoice), rather than qualitative deadlines which express an arbitrary property of a state (e.g., the seller should hand out the receipt before the buyer leaves the store). Studying norms with real-time deadlines is important because increasingly normative programming frameworks like [14] suggest the use of real-time deadlines. Several papers already provide ways of modelling norms with real-time deadlines, for example in deontic logic [7], metric temporal logic [5] or defeasible logic [9, 8].

However, in order to provide a semantic foundation for computational models for norms with real-time deadlines, our starting point is quite different. We build on top of an abstract *real-time semantic framework*, which takes timed state sequences as the basic semantic structure. We choose this structure because of its close correspondence with existing computational normative agent frameworks such as [6, 17, 13]. In this way our framework can form the semantic foundation of (real-time extensions of) these computational normative frameworks.

Moreover, in contrast with other work on norms with real-time deadlines, we identify the phenomenon of *deadline shifting*. We argue that deadline shifting should be taken into account when defining semantics for norms with real-time deadlines. Consider for example a norm that if the alarm rings, people are obligated to leave the building within 10 minutes. Now assume that the alarm keeps ringing. Intuitively, this should not mean that the deadline keeps shifting as well. Now consider that instead of an alarm ringing, a person broadcasts via a speaker that everyone has to leave a building within 10 minutes and repeats this announcement in 2 minutes. Then, arguably, the deadline shifts to 2 minutes later. We propose a semantic framework for norms with real-time deadlines that can distinguish these cases of (potential) deadline shifting.

The approach we take is the following. As in [3], we use a representation of a normative system that distinguishes between norms which exist in the social environment of agents and are 'in force' over extended periods of time, and the *detached obligations* which arise at a particular moment in time as the result of a norm becoming applicable. Explicit modelling of detachment in the framework

Appears in: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013), Ito, Jonker, Gini, and Shehory (eds.), May, 6–10, 2013, Saint Paul, Minnesota, USA.*

is important as deadline shifting is connected with repeated detachment. Moreover, we introduce explicit relations to model the two different kinds of norm interaction related to deadline shifting. We use the framework of timed transition systems [10] as the basis of our semantic framework (Section 4).

We extend timed transition systems by adding labellings to express which obligations and violations hold in the states of the system. The resulting models admit arbitrary obligation and violation labellings. We investigate semantics of norms with real-time deadlines declaratively by introducing several *constraints* on these labellings to ensure that these correspond to intuitions about semantics of norms (Sections 6 and 7). This allows us to study these individually and in combination as elements of the semantic framework, by means of which we can make explicit which constraints are sufficient to ensure certain properties of the framework. Through this study, we obtain a fundamental understanding of the semantics of norms with real-time deadlines and the related phenomenon of deadline shifting, which in turn can be used as a solid formal foundation for developing real-time normative systems. We conclude the paper in Section 8.

2. RELATED WORK

In this section we discuss related work in the area of norms with deadlines. Our framework is inspired in various ways by [3]. We use a similar representation of norms, namely as triples that consist of a triggering condition, a condition that represents an achievement obligation and a deadline. Also we define the semantics of norms as obligation and violation labellings on states. The main difference is that in [3] a framework for norms with qualitative deadlines (which refer to states) is defined, while we define a framework for norms with real-time deadlines. The latter gives rise to specific considerations that need to be taken into account explicitly in the framework: time has specific properties (e.g., it eventually progresses beyond any given time point) which has to be modelled explicitly to yield an appropriate semantic framework for norms with real-time deadlines; moreover, a framework for real-time deadlines needs to address the issue of deadline shifting as explained in the introduction. These aspects cannot be modelled and investigated in the qualitative framework of [3].

In [7] a deontic branching time temporal logic with a Kripke semantics is used that incorporates real-time aspects. The issue of deadline shifting is hinted at by stating that a deadline should only become active the first time that the triggering condition becomes true and providing a logical formula to model this. However, the authors do not identify the dual case where deadline shifting *is* desired, and consequently do not present a normative model that makes these different relation between norms explicit and analyze corresponding semantics, as we do.

In [5], monitoring of social expectations is investigated where these are expressed in a variant of metric interval temporal logic. The focus of the paper is not on investigating semantics for norms with real-time deadlines. Rather, the aim is to define a language for expressing social expectations that is amenable to monitoring. Consequently, real-time aspects such as deadline shifting are not addressed. The paper also does not separate norms and the obligations that they detach, as we do in this paper.

In [2] an operational semantics for timed normative multi-agent systems is proposed that is based on timed automata. In that work norms with real-time deadlines are not first-class citizens in the framework (instead they are modelled using clocks and clock constraints), and there is no explicit separation between norms and detached obligations. These aspects are important in our framework and allow a detailed investigation of time-related aspects, in particular investigation of deadline shifting and an accompanying analysis of how this affects detachment and termination of obligations.

3. NORMATIVE MODEL

Our work focuses on the class of norms that have associated real-time deadlines. In practice, norms that obligate an agent to achieve a particular state need to have deadlines in order to ensure that the accomplishment of the obligation is not delayed endlessly. Otherwise there is no point at which an agent can be said to have violated the norm and agents may not be motivated to fulfill their obligations [15]. Examples include payment obligations, delivering goods, responding to invitations (in case the agent intends to attend), and handing in an assignment (see [8] for a systematic classification).

3.1 Norms

In order to represent norms we assume a suitable language $\mathcal{L}$ for expressing conditions on states. For a state s and formula $\phi \in \mathcal{L}$, we write $s \models \phi$ ($s \not\models \phi$) to represent that ϕ holds (resp., does not hold) in state s. The only property of $\models$ that we need in our framework is that we cannot have both $s \models \phi$ and $s \not\models \phi$, which would give rise to conflicting obligation labellings.

We represent norms as triples of the form (c, ϕ, ttf), which informally means that if condition c holds *now*, then it becomes obligatory to achieve ϕ before the time-to-fulfill *ttf* has passed. The parameter *ttf* represents the time available to fulfill the obligation and indicates that ϕ must be achieved before the deadline $\mathit{now} + \mathit{ttf}$. This representation of norms thus gives rise to deadlines that are *relative* to the time at which the triggering condition of the norm holds. For example, the norm $(\mathit{pay}_{bs}, \mathit{receipt}_{sb}, 1)$ means that if the buyer has paid the seller *now*, then the seller is obligated to send a receipt to the buyer within 1 day, i.e., before time $\mathit{now} + 1$ (example adapted from [3]). In this paper our main concern is with norms with relative deadlines, which is the more interesting case, but norms with absolute deadlines can straightforwardly be added if needed. Norms are formally defined as follows.

DEFINITION 1. *(Norm) A* norm *is a triple* (c, ϕ, ttf) *with* $c, \phi \in \mathcal{L}$ *and* $\mathit{ttf} \in \mathbb{R}_0^+$, *i.e. ttf is a positive real number or zero.*

We allow zero time-to-fulfill but do not allow negative time-to-fulfill. Norms with zero time-to-fulfill require immediate fulfillment of an obligation. This allows, for example, to express that a hotel employee should see to it that the hotel entrance is unlocked when a client is at the entrance without any delay.

Throughout the paper we assume that a set of norms $\mathcal{N}$ has been fixed. We thus assume that norms are a stable part of an agent's social or organizational environment (cf. [4]).

Norms give rise to obligations and violations. The condition c of a norm (c, ϕ, ttf) acts as a trigger for the creation of obligation ϕ, and *ttf* is used to set a deadline relative to the time that the trigger holds. If the triggering condition of a norm (c, ϕ, ttf) holds and an obligation to achieve ϕ is created, we say that the obligation is *detached* by the norm. If ϕ is not reached before the deadline, a violation occurs. Obligations and violations thus originate from norms at a particular time and for particular situations.

3.2 Normative Systems

Norms with real-time relative deadlines give rise to particular issues that need to be addressed in their operationalization. These are related to (potential) *deadline shifting* due to repeated detachment of the same obligation. To illustrate this, consider again the example of a norm saying that if an alarm rings, everyone is obligated to leave a building within 10 minutes. Assume that the alarm keeps ringing. In this case, the intended meaning is not that new obligations are detached continuously, which would mean that the deadline keeps shifting. Rather, the first application of the norm should *block* consecutive applications in order to prevent detachment of new obligations.

Whether one application of a norm should block consecutive applications is domain dependent. There are also examples where deadline shifting, either to an earlier or later time, *is* desirable. If, for example, instead of an alarm a person broadcasts via a speaker that everyone has to leave a building within 10 minutes and repeats this announcement in 2 minutes, arguably, the deadline shifts to 2 minutes later. In this case, application of a norm *cancels* obligations that were already detached by other norms (or earlier applications of the same norm).

In order to model these different interactions between norms, we introduce two binary relations. The fact that one norm may block the application of another norm is modelled through a binary *blocking relation* **B** on norms. Informally, $n\mathbf{B}n'$ means that the norm n that has been applied earlier (or possibly in the same state) *blocks* the application of norm n'. This allows us to model the alarm example by using the **B** relation to ensure that the norm to leave the building blocks later applications of that same norm. Although the particular example concerns blocking of a norm by the very same norm that was applied earlier, by introducing a blocking relation our model also is able to account for the more general case where any norm may block another norm.

A second binary *cancelling relation* $n\mathbf{C}n'$ is introduced to model cancellation of one norm by another. Informally, $n\mathbf{C}n'$ means that norm n *cancels* norm n' that has been applied earlier (or in the same state). This allows us to model the broadcasting example. In addition, it allows to model cases that do not concern deadline shifting but that do require cancellation of obligations. For example, the obligation to finish a paper before the deadline is canceled by an obligation to prepare a lecture in time.

Besides the issue of deadline shifting which is specific to norms with real-time relative deadlines, the framework also needs to address persistence of obligations, which is relevant for norms with any kind of deadline (see also [8]). Intuitively, obligations persist at least until the deadline (if they are not fulfilled). If at the deadline, however, an obligation is not fulfilled, it may either persist or terminate. For example, an obligation to pay typically persists but an obligation to deliver a wedding cake does not persist when the wedding has ended [8]. In the latter case the obligation terminates at the deadline whereas in the former it does not.

In order to distinguish between these cases we introduce a set **D** that contains all norms whose associated obligations terminate at the deadline. We say that such a norm satisfies *deadline termination*. The complement $\mathcal{N} \setminus \mathbf{D}$ consists of all those norms whose associated obligations do not terminate at the deadline.

We formally define a normative system to consist of a set of norms, the blocking and cancellation relations and the set of norms that terminate at the deadline.

DEFINITION 2. *(Normative System) A* normative system *NS is a tuple* $(\mathcal{N}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$ *where* $\mathcal{N}$ *is a set of norms,* $\boldsymbol{B}$ *is an a-cyclic binary relation on the set of norms (expressing blocking),* $\boldsymbol{C}$ *is an a-cyclic binary relation on the set of norms (expressing cancellation), and* $\boldsymbol{D}$ *is the subset of norms that satisfy deadline termination.*[1]

The requirement that the blocking and cancelling relations are a-cyclic in particular prevents that two norms mutually block or cancel each other, and, in the more general case, that there is a cycle of norms which all in turn block or cancel each other. Both relations may be reflexive to allow for the case where a norm blocks or cancels itself if it is triggered in consecutive states (which is the case in both the alarm and the broadcasting example above).

It is important to note that the triggering condition of norms in our framework, in line with other frameworks like [3, 13], expresses a condition on a state rather than a condition on the occurrence of an action or event. The latter can also be expressed by conditions on a state if the state represents the actions that have been executed or events that have occurred, and conditions on state may be used to represent the more general case. This is important because blocking can be desirable in cases where the triggering condition continues to hold, which will in particular be the case if this condition refers to a property of the state, such as the alarm ringing. If the condition refers to the action that caused the alarm to start ringing or to the event of the alarm starting to ring, the condition will not continue to hold and thus blocking may not come into play. We argue for the more general representation of norms here in which the triggering condition refers to a state rather than (only) to an action or event, and therefore explicitly consider and model the case of blocking norms.

4. REAL-TIME SEMANTIC FRAMEWORK

For our purposes, it is important to build on top of existing computational semantics for multi-agent systems. As it has been common to provide semantics for multi-agent systems using an operational, interleaving semantics, we have chosen to built our semantics for normative systems on top of a well-known real-time framework called *timed transition systems*. The traditional transition systems semantics is abstract with respect to time and cannot be used for modelling real-time systems in which one wants to keep track of the time at which actions are performed [11]. Timed transition systems provide a conservative extension of traditional transition systems semantics that has been commonly used to provide a formal semantics for multi-agent systems and computational models of organizations as these are inherently concurrent, see e.g. [6, 17].

Timed transition systems extend the traditional interleaving model by incorporating time and thus allow for the analysis of real-time systems. We use the framework of timed transition systems as introduced in [10] as our basic semantic framework. A transition system consists of a set of states Σ, some of which are initial states in which a computation can start, and a finite set of transitions that indicate which state changes can occur in the system. Transitions model system behavior and the basic actions that can be performed by agents. An infinite sequence of states where consecutive state pairs are transitions is called a *computation*. Timed state sequences add a corresponding sequence of times to a state sequence.

DEFINITION 3. *(Timed State Sequence) Let* Σ *be a set of states. A timed state sequence* $\rho = (\sigma, \boldsymbol{T})$ *consists of an infinite sequence* σ *of states* $s_i \in \Sigma$ *with* $i \geq 0$*, and an infinite sequence* $\boldsymbol{T}$ *of corresponding times* $t_i \in \mathbb{R}$ *with* $t_i \geq 0$*.*

In line with our general approach of introducing constraints to ensure the adequacy of our semantic model, two constraints are imposed on the basic real-time semantic framework. The first constraint enforces that time monotonically increases and, in particular, never decreases (weak monotonicity) as expected.

CONSTRAINT 1. *[Monotonicity, [10]] A sequence* ρ *satisfies* monotonicity *if for all* $i \geq 0$*, either* $t_{i+1} = t_i$*, or* $t_{i+1} > t_i$ *and* $s_{i+1} = s_i$*.*

[1] A binary relation R is cyclic if there are $n_1, \ldots, n_k$, with $k > 1$, such that $n_i R n_{i+1}$ for $1 \leq i \leq k-1$ and $n_k R n_1$ and there is an i such that $n_i \neq n_{i+1}$; otherwise it is a-cyclic.

Note that the monotonicity constraint does not only constrain the progress of time but also requires that states do not change if time progresses and vice versa. This is in line with the traditional interleaving model which abstracts from the fact that actions take time. This abstraction allows for the arbitrary interleaving ("shuffling") of transitions [12]. Building on this framework, time is incorporated into the timed transition system model by assuming that all transitions happen "instantaneously", while real-time constraints restrict the times at which transitions may occur [10]. In summary, timed transition systems distinguish between two kinds of transitions: transitions which represent *state activities* in which the state may change but time does not advance and transitions that represent *time activities* in which time changes but states do not.

The second constraint that is imposed ensures that time is *divergent*, i.e. eventually progresses such that for arbitrary $t \in \mathbb{R}$ we can always find a time $t_i > t$. Note that monotonicity is insufficient to guarantee progress because time sequences may converge. Progress is an essential constraint in any adequate model of norms with real-time deadlines because a deadline might never occur if time does not make sufficient progress.

CONSTRAINT 2. *[Progress, [10]] A timed state sequence ρ satisfies* progress *if for all $t \in \mathbb{R}$ there is an $i \geq 0$ such that $t_i > t$.*

Because the time domain $\mathbb{R}$ does not have a maximal element, progress also implies that there are infinitely many time steps in a sequence. It follows that timed state sequences alternate state activities and time steps. This also means that in any time interval only *finitely* many state changes can occur and that timed state sequences satisfy *finite variability*. As is usual, we will call a subsequence of a timed state sequence that only consists of state activities a *micro-phase* and a subsequence that only consists of time steps a *macro-phase* [1].

5. NORM-BASED LABELLINGS

Norms give rise to obligations which in turn can be violated. The idea is that detachment of a norm gives rise to an obligation, and if the obligation is not achieved before the deadline, a violation occurs. In order to provide a semantics for norms, the states in timed state sequences are *labelled* with information about the obligations that are active and about violations that have occurred. We use two labellings: one that indicates which obligations hold and one which indicates which violations occurred in a state of a timed state sequence. In order to define such labellings, however, it turns out to be more convenient to first define labellings that keep track of the norm itself that gives rise to an obligation or violation in combination with the associated deadline.

DEFINITION 4. *(Labelled Timed State Sequence) Let NS be a normative system with set of norms $\mathcal{N}$. A* labelled timed state sequence $(\rho, O^{NS}_\rho, \mathcal{V}^{NS}_\rho)$ *consists of a timed state sequence ρ, and mappings $O^{NS}_\rho : \mathbb{N} \rightarrow \wp(\mathcal{N} \times \mathbb{R})$ and $\mathcal{V}^{NS}_\rho : \mathbb{N} \rightarrow \wp(\mathcal{N} \times \mathbb{R})$. We call O^{NS}_ρ a norm-based obligation labelling and $\mathcal{V}^{NS}_\rho$ a norm-based violation labelling for ρ.*

Informally, $(n, d) \in O^{NS}_\rho(i)$ with $n = (c, \phi, \mathit{ttf})$ means that it is obligatory to achieve ϕ before deadline d, and that ϕ was detached by n. $(n, d) \in \mathcal{V}^{NS}_\rho(i)$ with $n = (c, \phi, \mathit{ttf})$ means that the obligation to achieve ϕ before deadline d was violated, and that ϕ was detached by n. In the remainder we also use ρ to refer to labelled timed state sequences, write $n \in O^{NS}_\rho(i)$ to denote that there is an $(n, d) \in O^{NS}_\rho(i)$, and use $n \in \mathcal{V}^{NS}_\rho(i)$ similarly. It is clear that from a given norm-based labelling we can extract a labelling that associates only obligation-deadline pairs with a state.

DEFINITION 5. *(Abstract Norm-based Labelling) Given a labelling L^{NS}_ρ with $L \in \{O, V\}$, the* abstract labelling *L_ρ is defined by: $(\phi, d) \in L_\rho(i)$ iff there is a norm $(n, d) \in L^{NS}_\rho(i)$.*

The abstract representation (ϕ, d) of a norm combined with a deadline represents the information needed to detect violations. It also is sufficient for handling different, but overlapping windows for fulfilling one and the same obligation. For example, an agent may be obliged to pay a 50 Euro fine within 2 weeks while the (unfortunate) agent receives a second fine before the end of these 2 weeks that obligates him *again* to pay 50 Euro for a possible other offence. The representation of norm-deadline pairs is expressive enough to represent such examples.

We write $\phi \in O_\rho(s)$ whenever $(\phi, d) \in O_\rho(s)$ for some d, and similarly for $\mathcal{V}$. Note that violations are represented as pairs (ϕ, d) to keep track of the specific obligation and associated deadline that has been violated. This is useful because choosing an appropriate sanction typically depends on the obligation and the time that has passed since the deadline. For example, one might want to impose a sanction when a student assignment has been handed in too late that reduces the score depending on the number of late days involved.

We use Ψ_{chaos} to denote the set of all labelled timed state sequences that satisfy the two constraints introduced in Section 4. This set contains all possible labellings of a timed state sequence. In the remainder, we introduce constraints on this set that restrict this set to the set of adequate labellings.

6. OBLIGATION LABELLINGS

Now that we have put the semantic framework in place, we proceed by defining which labellings of states are adequate. By adequate labellings we mean labellings that correspond to basic *intuitions* about the behavior of norms with deadlines (as also reflected in other frameworks such as [14, 8]), as well as those that lead to a *well-defined* semantics of norms. The semantics should correspond to intuitions concerning detachment, termination and persistence of obligations, in such a way that the real-time aspect of norms is taken into account. The semantics is well-defined if it results in a unique labelling for each timed state sequence, meaning that it is consistent (it defines at least one labelling) as well as coherent (it specifies no more than one labelling).

Our approach is to introduce constraints on the set Ψ_{chaos} of labelled timed state sequences to yield an adequate labelling. This allows us to study these constraints individually and in combination as elements of the semantic framework. In this section we investigate the concepts and constraints that result in adequate obligation labellings, and in Section 7 we build on top of this and introduce a constraint to obtain adequate violation labellings. We define obligation labelings with respect to a normative system $NS = (\mathcal{N}, \mathbf{B}, \mathbf{C}, \mathbf{D})$.

6.1 Detachment of an obligation

The basic idea is that an obligation is detached when a norm is applicable. A norm (c, ϕ, ttf) is only applicable in a state if its trigger c holds. The obligation ϕ should not be detached in a state, however, if ϕ already holds in that state, because it makes little sense to apply a norm whose obligation is already fulfilled.

DEFINITION 6. *(Applicable) Let s be a state. A norm $n = (c, \phi, \mathit{ttf})$ is* applicable *in state s if it is triggered in state s, i.e. $s \models c$, and the obligation ϕ is not (yet) fulfilled, i.e. $s \not\models \phi$.*

The detachment of an obligation by a norm, however, not only depends on the applicability of a norm but also on interactions of that norm with other norms that may prevent detachment. In particular, an active norm n that has already been applied before can block the application of norm n' in i (that is, if $n\mathbf{B}n'$). We say that a norm is *active* in state i if $n \in O_\rho^{NS}(i)$. Moreover, if different norms n, n' for which $n\mathbf{B}n'$ are triggered in the same state (and have not been triggered before), n should block detachment of n'. This is reflected in the definition of what it means that a norm blocks another in a particular state.

DEFINITION 7. *(Blocking) Let O_ρ^{NS} be a labelling for ρ, and $\boldsymbol{B}$ be a blocking relation. We say that norm n* blocks *(the application of) norm n' in state i if n is active in i and $n\boldsymbol{B}n'$. In that case, we also say that norm n'* is blocked *in state i.*

Note that blocking is defined relative to a given, arbitrary labelling. Blocking thus is also defined for labellings that are inadequate. This is in line with our approach of filtering out or eliminating "bad" labellings by means of the constraints we introduce below. Of course, it is in particular important that the definition is correct for adequate labellings. We return to this issue at the end of this section. Also note that two different norms cannot block each other because **B** is a-cyclic.

A norm n that is applicable in a state may cancel the application of a norm n' that is about to be applied, but only if n is not blocked. Blocking thus takes precedence over cancelling. The cancelling relation **C** then indicates whether a norm actually cancels a norm. Note that two different norms cannot cancel each other because **C** is a-cyclic.

DEFINITION 8. *(Cancelling) Let O_ρ^{NS} be a labelling for ρ, and $\boldsymbol{C}$ be a cancelling relation. We say that norm n* cancels *norm n' in state i if n is applicable but not blocked in i, $n \in O_\rho^{NS}(i)$, and $n\boldsymbol{C}n'$. In that case, we also say that norm n' is cancelled in state i.*

Detachment or activation of an obligation ϕ in a state s_i by applying norm n should lead to an obligation labelling in that state that includes ϕ and associated absolute deadline d which can be computed by adding *ttf* to the time of the state in which the norm is applied. This is implemented by the following constraint, which takes blocking and cancelling into account and represents a general principle that should be satisfied by any adequate labelling.

CONSTRAINT 3. *[Detachment] A labelling O_ρ^{NS} satisfies the* detachment condition *if for all $n = (c, \phi, ttf) \in \mathcal{N}$: if n is applicable but not blocked nor cancelled in i, then $(n, t_i + ttf) \in O_\rho^{NS}(i)$.*

We now return to the discussion of the correctness of our definition of blocking, and in particular the case in which a norm blocks itself, i.e., $n\mathbf{B}n$. In this case, the intended semantics is that if n is applicable in state i (and not blocked by another norm), it should be the case that $(n, t_i + ttf) \in O_\rho^{NS}(i)$. Labellings that do not satisfy that $n \in O_\rho^{NS}(i)$ are excluded according to Definition 3, as in that case n would not be blocked and thus it should be in the labelling. Labellings that satisfy $(n, t_i + ttf) \in O_\rho^{NS}(i)$ are allowed by this definition, but it also allows labellings that satisfy $(n, d) \in O_\rho^{NS}(i)$ where $d \neq t_i + ttf$. Filtering out such labellings with erroneous deadlines is done through Constraints 8 and 9 as introduced below.

6.2 Termination of an obligation

Obligations may be terminated for several reasons. We consider the following reasons in this paper. The most obvious reason is that an agent has *fulfilled an obligation*, whether or not this is achieved before or after the deadline. This type of termination represents a general principle that applies to all norms that detach *achievement obligations* whether with or without real-time deadlines. Second, an obligation may also be terminated because *the deadline* associated with the obligation *has passed*. For example, the obligation to deliver a good (e.g. a wedding cake) is terminated because it is of no use any more to deliver the good after the deadline [8]. This type of termination, however, does not always apply as an obligation to pay, for example, typically persists also after the deadline has passed. Third, an obligation may be terminated because one norm cancels the application of another norm (see also Section 3).

Fulfillment termination can be implemented by imposing a simple constraint that says that if ϕ is achieved in a state, it cannot be an obligation in that state. Intuitively, *achievement* obligations are only associated with states where the obligation still needs to be achieved but not with those where the obligation has been realized. The constraint is imposed on an abstract labelling O_ρ which has implications for the corresponding labelling O_ρ^{NS}. That is, if a constraint requires $(\phi, d) \notin O_\rho(i)$, by Definition 5, it follows we cannot have that $(n, d) \in O_\rho^{NS}(i)$ for $n = (c, \phi, ttf)$.

CONSTRAINT 4. *[Fulfillment termination] An abstract labelling O_ρ satisfies* fulfillment termination *if for all ϕ and $i \geq 0$: if $s_i \models \phi$, then $(\phi, d) \notin O_\rho(i)$ for any d.*

As discussed above, obligations may or may not be terminated when a deadline is reached, depending on the norm that detached the obligation. The set **D** of a normative system contains the norms whose associated obligations should terminate at the deadline. Note that an obligation is not terminated at the deadline itself but only after the deadline has passed.

CONSTRAINT 5. *[Deadline termination] A labelling O_ρ^{NS} satisfies* deadline termination *if for all $n \in \mathcal{N}$ and $i \geq 0$: if $(n, d) \in O_\rho^{NS}(i)$, $n \in \boldsymbol{D}$, and $t_{i+1} > d$ then $(n, d) \notin O_\rho^{NS}(i + 1)$.*

The next constraint enforces the removal of norms that are cancelled by an applicable norm. That is, if a norm n is applicable in i and cancels another norm n' that has been applied, the norm n' should be removed from the obligation labelling associated with i. The cancelling relation **C** indicates which norms cancel each other.

CONSTRAINT 6. *[Cancelling] A labelling O_ρ^{NS} satisfies* cancellation *if for all $n = (c, \phi, ttf), n' \in \mathcal{N}$ and $i \geq 0$: if n is applicable but not blocked in i and $n\boldsymbol{C}n'$, then $(n', d) \notin O_\rho^{NS}(i) \backslash \{(n, t_i + ttf)\}$ for any d.*

As for blocking, we need to take some care here that one and the same norm does not completely cancels its own application in a particular state while we still allow the same norm to cancel its own earlier applications. This explains the particular condition in Constraint 6 which does not impose any requirements for a single norm n as $(n, t_i + ttf) \notin O_\rho^{NS}(i) \setminus \{(n, t_i + ttf)\}$ is trivially true. For suppose that n is applicable, not blocked, $n\mathbf{C}n$, and we would have $n \notin O_\rho^{NS}(i)$. According to Definition 8, in that case n does not cancel itself and by Constraint 3 we must have $n \in O_\rho^{NS}(i)$ and arrive at a contradiction. Because Constraint 6 allows that $n \in O_\rho^{NS}(i)$, we get $n \in O_\rho^{NS}(i)$ as desired.

Note that Constraint 6 does allow self-cancellation in case the same norm has been applied before. For example, a school teacher that obligates children to be back in the classroom within 10 minutes and renews that same obligation at some later time intuitively cancels the earlier obligation. Because the deadline shifts in this case and we have $d \neq t_i + ttf$ the cancellation constraint takes care of such cases and would indeed remove obligations earlier introduced by one and the same norm.

Constraint 6 ensures that a norm n that is cancelled at i is removed from the labelling, i.e. labellings that associate n with i are ruled out. It does not rule out the *continuous shifting* of a deadline, however. Even in the simple example of the school teacher above this is problematic because it would render issuing the obligation rather meaningless. By continuously shifting a deadline, in effect no real deadline is set. Accomplishing the obligation could thus be delayed endlessly, which is undesirable, as discussed in Section 3.

In our setting, continuous shifting of a deadline may occur, for example, if a norm $n = (c, \phi, ttf)$ cancels another norm $n' = (c', \phi, ttf')$ that both detach obligation ϕ. Suppose, for example, that we have $(n', d') \in O^{NS}_\rho(i)$ and that n' is cancelled by n at $i+1$. In effect, (n', d') then is removed and (n, d) with $d = t_{i+1} + ttf$ is added to the labelling of $i + 1$. As a result, the deadline associated with obligation ϕ shifts in case $d \neq d'$. Such shifts may happen more than once, resulting in a chain of repeated shifts of the deadline associated with ϕ. In order to rule out infinite chains of deadline shifts, we first formally define such chains.

DEFINITION 9. *(Chain of Deadline Shifts) Let O^{NS}_ρ be a labelling. A sequence of norms $n_0 = (c_0, \phi, ttf_0), n_1 = (c_1, \phi, ttf_1), n_2 = (c_2, \phi, ttf_2), \ldots$ is a* chain of deadline shifts for ϕ on ρ *iff there is a sequence of increasing indices $i_0, i_1, i_2, \ldots$ such that for all $k \geq 0$:*

- *for all $i_k \leq j < i_{k+1}$: $n_k \in O^{NS}_\rho(j)$, and*
- *n_{k+1} cancels n_k in i_{k+1}.*

A maximal *chain of deadline shifts is a chain that cannot be extended into a longer chain.*

To avoid infinite chains of deadline shifts, we introduce the following constraint.

CONSTRAINT 7. *[Termination of Chains of Deadline Shifts] All maximal chains of deadline shifts are finite.*

6.3 Persistence of obligations and terminations

The activation as well as termination of obligations should persist but this is not guaranteed by the constraints introduced above. These constraints do, however, introduce *stable obligations* in the labelling of state sequences. The Detachment Constraint 3, for example, requires that all labellings satisfy $\phi \in O_\rho(i)$ if a norm is applicable and is not blocked nor cancelled in i. Similarly, termination constraints require the absence of an obligation in the labeling of a state. We formally define the stability of an obligation relative to a *set* of labeled state sequences.

DEFINITION 10. *(Stable Obligation at Index) Let Ψ be a set of arbitrary labeled state sequences. Then ϕ is a* stable obligation at *i iff for any two labeled state sequences $(\rho, O, \mathcal{V}), (\rho, O', \mathcal{V}') \in \Psi$, we have that $\phi \in O(i)$ iff $\phi \in O'(i)$. We also say that i is a* stable index for ϕ. *Otherwise ϕ is un*stable at i. *Finally, an index i is a* maximal stable index for ϕ with respect to *j if ϕ is stable at i, $i < j$, and there is no $i < k < j$ such that ϕ is stable at k.*

Intuitively, the idea is that the presence or absence of a *stable* obligation should persist up and until the next stable labelling for that obligation. The persistence constraint is defined relative to a given *set* of labellings. Note that the two constraints we introduce here are different in that respect from those before because we can only determine stable obligations with respect to a set. The constraint ensures that obligations do not miraculously (dis)appear, which is reasonable if we assume that obligations can only be introduced by (publicly available) norms.

CONSTRAINT 8. *[Persistence] Let Ψ be a set of arbitrary labeled state sequences. The persistent subset $\Psi_{persist} \subseteq \Psi$ is the set of all labeled state sequences $\rho \in \Psi$ that satisfy for all obligations $\phi \in \mathcal{L}$: if ϕ is unstable at j and $i < j$ is the maximal stable index for ϕ, then $\phi \in O_\rho(j)$ iff $\phi \in O_\rho(i)$.*

Persistence does not guarantee a stable labeling at the first state of a sequence. The next constraint ensures that only those obligations are associated with an initial state that are stable at the initial state (index 0).

CONSTRAINT 9. *[Initial States] Let Ψ be a set of labeled state sequences. The initial subset $\Psi_{init} \subseteq \Psi$ is the set of all labelled sequences $\rho \in \Psi$ that satisfy for all obligations $\phi \in \mathcal{L}$: if ϕ is unstable at 0, then $\phi \notin O_\rho(0)$.*

6.4 Properties

We will now show that certain classes of labellings determine a unique labelling for a state sequence. We write Ψ_{constr} to denote the subset of labelled state sequences in Ψ that satisfies constraint $constr$. We refer to constraints by single letters: a refers to detachment (Constr, 3), f to fulfillment termination (Constr. 4), d to deadline termination (Constr. 5), c to cancellation (Constr. 6), t to termination of deadline shifts (Constr. 7), p to persistence (Constr. 8), and i to initialization (Constr. 9). For example, Ψ_a is that subset of Ψ that satisfies the Detachment Constraint. We also write Ψ_{c1+c2} to denote $\Psi_{c1} \cap \Psi_{c2}$. For example, Ψ_{a+p+i} is the set of labelled sequences ρ that satisfy the Detachment Constraint, are such that stable obligation labellings persist, and have stable obligation labellings for the initial state.

We say that the obligation labelling of a timed state sequence ρ is *unique* in a set of labeled sequences Ψ if for all $(\rho, O, \mathcal{V})$, $(\rho, O', \mathcal{V}') \in \Psi$ we have that $O = O'$. It is clear that labellings that do not satisfy the Initialization or the Persistence Constraint are not necessarily unique; for example, Ψ_{a+p} and Ψ_{a+i} need not be unique. Also note that in general by adding a constraint we do not necessarily obtain a subset of labellings. For example, $\Psi_{a+p+i} \not\subseteq \Psi_{a+f+d+p+i}$ since in the former set norms are never terminated. For the following we can prove existence and uniqueness.

THEOREM 1. *(Unique Labellings) Let Ψ be the set of all labeled state sequences, i.e. $\Psi = \Psi_{chaos}$. Then the following sets of labeled state sequences are unique and non-empty: Ψ_{a+p+i}, $\Psi_{a+f+p+i}$, $\Psi_{a+d+p+i}$, $\Psi_{a+f+d+p+i}$, $\Psi_{a+f+c+d+t+p+i}$.*

Proof: We provide an outline of the proof. We need to show that for any two labellings $O_\rho, O'_\rho \in \Psi_{\ldots}$ and for any of the subscripts substituted for ... we have that $O_\rho = O'_\rho$. That is, the labeling for ρ is unique. Most importantly, we need to show that the initial state is uniquely defined. Given that both the blocking and cancellation relation are a-cyclic, we need to distinguish two cases: either we can find a norm that is not blocked nor canceled itself by any norm, or we have an infinite sequence of norms that block or cancel each other. In the later case, by the Detachment Constraint 3 (subscript a*), none of these norms can detach an obligation and by the Initialization Constraint 9 (*i*) we then obtain that the associated obligations are absent in the labeling. For the other case, Constraint 3 is sufficient to show the obligation must be present in the labeling. Now, suppose further, to arrive at a contradiction, that $O_\rho(i) \neq O'_\rho(i)$, for some index i. In that case, the obligations associated with state i are not stable and there is an obligation $(\phi, d) \in O_\rho(i)$ and $(\phi, d) \notin O'_\rho(i)$, or vice versa. Again we need to consider two cases: either there is a relevant constraint that rules out that (ϕ, d) is or is not associated with state i, or not.*

In the former case, using one of the relevant Constraints 3,4, 5, or 6 associated with each of the different sets of labellings $\Psi_{\ldots}$ is sufficient to arrive at a contradiction. In the other case, use the Persistence Constraint 8 to get a contradiction. $\square$

A basic intuition related to an achievement obligation ϕ is that if it has been fulfilled it is no longer an obligation. Proposition 1 shows that Fulfilment Termination (Constr. 4) and Persistence (Constr. 8) are sufficient to prove this property and shows that in the presence of Detachment (Constr. 3) we must additionally require that in the meantime no norm which detaches ϕ is applicable.

PROPOSITION 1. *(Achievement Obligations) Let $\Psi = \Psi_{chaos}$.*

1. *Let $\rho \in \Psi_{f+p}$. If ϕ is achieved in state i, i.e. $s_i \models \phi$, then $\phi \notin O_\rho(j)$ for $j \geq i$.*
2. *Let $\rho \in \Psi_{a+f+p}$. If ϕ is achieved in state i, and for all $i \leq k \leq j$ no norm n which detaches ϕ is applicable in k, then $\phi \notin O_\rho(j)$.*

Proof: We sketch the proof for (1). Suppose $s_i \models \phi$. According to Constraint 4, ϕ is stable at i and $\phi \notin O_\rho(i)$. Note that ϕ can only be stable at k relative to Ψ_f if $\phi \notin O_\rho(k)$ for arbitrary $(\rho, O, V) \in \Psi_f$. So, if ϕ is not stable for a $j > i$, it follows by Constraint 8 that we must have $\phi \notin O'_\rho(j)$ for all $(\rho, O', V) \in \Psi_{f+p}$. $\square$

Vice versa, if an obligation has been terminated, intuitively it must be the case that the obligation has been fulfilled. That is, unless an obligation has been achieved (or is terminated after its deadline in line with Constraint 5), it should persist.

PROPOSITION 2. *(Persistence of Obligations) Let $\Psi = \Psi_{chaos}$.*

1. *Let $\rho \in \Psi_{f+p}$. If ϕ is stable at j, then $\exists i < j$ such that $s_i \models \phi$.*
2. *Let $\rho \in \Psi_{f+d+p}$. If ϕ is stable at j, then $\exists i < j$ such that $s_i \models \phi$ or a deadline d for ϕ has been passed at j ($t_j > d$).*

Proof: Observe that ϕ can only be stable at j relative to Φ_{f+p} or Φ_{f+d+p} if $\phi \notin O_\rho(j)$. Use Constraint 5 for (2). $\square$

7. VIOLATION LABELLINGS

The intuitive idea of an adequate violation labelling is that a violation occurs if an obligation is not achieved before the deadline. In our framework this translates into the requirement that whenever (n, d) is associated with a state i by an obligation labelling $O^{NS}_\rho(i)$ and the deadline $d < t_i$ has passed, (n, d) should also be in the violation labelling. $(n, d) \in \mathcal{V}^{NS}(i)$ thus represents that the obligation detached by n has not been achieved by d. As this remains true for all subsequent states, we also require that once (n, d) is in the violation labelling for a state all later states are similarly labelled.

CONSTRAINT 10. *[Violations] A labelling $\mathcal{V}^{NS}_\rho$ satisfies the* violation condition *if for all $i \geq 0$: if $(n, d) \in O^{NS}_\rho(i)$ and $t_{i+1} > d$ then $(n, d) \in \mathcal{V}^{NS}_\rho(i + 1)$, and if $(n, d) \in \mathcal{V}^{NS}_\rho(i)$, then $(n, d) \in \mathcal{V}^{NS}_\rho(i + 1)$.*

Depending on the amount of time that elapses between states in a macro phase, the triggering of a violation can be significantly later than the passing of the deadline (as the deadline can pass in between two states). A labelled timed state sequence can be *refined* to ensure that the passing of a deadline is detected within some reasonable time window. For reasons of space, we do not discuss this here but note that our framework can accommodate for such issues by requiring that the level of refinement is related to the specific *monitoring mechanism* of an organization in the context of which the norms are modelled. The maximum delay between states in a macro phase (see Section 4) in that case should correspond to the frequency with which this mechanism checks for violations.

Below we present two propositions that express properties of violations. The first investigates the case of obligations with 0 time left to fulfill the obligation. Intuitively, these are violated, unless the state to be achieved is achieved "immediately". In particular, norms with zero time to fulfill, which introduce such obligations, would lead to violation without immediate fulfillment. In our model of timed transitions systems, if there is zero time to fulfill in a state, the obligation needs to be realized within the corresponding micro-phase (as time does not progress in a micro-phase).

As before, we add a subscript v to a set of labelled state sequences to denote the subset of those labellings that satisfy Constraint 10.

PROPOSITION 3. *(Zero Time to Fulfill) Let $\Psi = \Psi_{chaos}$ and $\rho \in \Psi_v$. For all $i \geq 0$: if $(\phi, d) \in O_\rho(i)$ with $d = t_i$ and for all $j > i$ with $t_i = t_j$ we have $(\phi, d) \in O_\rho(i)$, then there is a $k > i$ such that $(\phi, d) \in \mathcal{V}_\rho(k)$.*

Proof: By the Progress Constraint 2, there is a smallest k with $t_k > t_i = d$. Then $t_{k-1} = t_i$ and it follows that $(\phi, d) \in O_\rho(k - 1)$. Using Constraint 10 we then must have $(\phi, d) \in \mathcal{V}_\rho(k)$. $\square$

Another basic intuition is that an obligation is either met or will be violated at some point in time. As a consequence, we would expect that in case fulfillment is never realized (i.e. Constraint 4 is not satisfied but the Persistence Constraint 8 is), we should be able to prove a violation will occur. There is one exception to this rule, however, due to the possible cancellation of a norm before its deadline. The next proposition shows that if we only allow *self-cancellation* our intuition is confirmed for various combinations of constraints and obligations are eventually violated if never fulfilled.

PROPOSITION 4. *(Violation Occurs Eventually)*

1. *Let $\rho \in \Psi_{a+v+i+p}$. If $\phi \in O_\rho(i)$, then there is a $j > i$ such that $\phi \in \mathcal{V}_\rho(j)$.*
2. *Let $\rho \in \Psi_{a+o+t+v+i+p}$ and $\mathbf{C} \subseteq \mathcal{N} \times \mathcal{N}$. If $\phi \in O_\rho(i)$, then there is a $j > i$ such that $\phi \in \mathcal{V}_\rho(j)$.*

Proof: We sketch the proof for (2) where we may suppose that Constraints 3, 6, 7, 8, 9 are satisfied and norms only cancel themselves. If an obligation would be terminated, it must be because of cancellation. But we only have self-cancellation, which means that only the deadline can have been shifted and by Constraint 7 this can happen only finitely often. Eventually the obligation thus must be violated. $\square$

8. CONCLUSION

We have proposed a semantic framework for norms with real-time deadlines and shown that it satisfies several desired properties. A normative system has been introduced that takes into account norm interactions where norms block and cancel each other's application. Such interactions arise naturally in a real-time setting but are also useful in other settings that do not involve real-time aspects explicitly. A labelling approach has been used to define the semantics for norms that detach achievement obligations and for formally specifying the violations of such norms. Several constraints have been introduced to eliminate labellings that are not adequate, taking into account the real-time aspects of the framework. Our approach based on explicitly introducing constraints allows for a precise study of various aspects, either separately or in combination, that need to be taken into account when handling norms with deadlines. In particular, we have studied deadline shifting which gives rise to particular issues that need to handled when specifying semantics of norms with relative real-time deadlines.

The semantic framework introduced builds on top of the well-known real-time semantic framework of timed transition systems. The advantage of using timed transition systems is that they are a conservative extension of the transition system semantics that is commonly used for formally specifying multi-agent systems. Our framework therefore contributes to this area by providing semantic foundations for specifying the semantics of norms with real-time deadlines in computational organizational systems.

An important topic for future work concerns the definition of an operational semantic framework that adheres to the constraints proposed here. This will form the basis for extending existing frameworks for organized multi-agent systems with real-time aspects [6, 17] and providing a formal foundation for, e.g., the normative programming framework with real-time deadlines of [14]. One topic that needs to be addressed in this regard concerns the development of an algorithm that is able to keep track of the norms and obligations that are active, and the detection of deadline violations. Whereas the main aim of our framework has been to provide a formal and well-defined semantics that conforms to basic intuitions about norms with deadlines, we also need to develop a mechanism for computing labellings while time progresses.

Our framework provides a basis for studying monitoring mechanisms and the granularity of time in more detail. In particular, monitoring raises new issues related to the detection of violations in a real-time setting. The notion of refinement may be particularly interesting in this context and we would like to study how various normative systems relate to each other when the granularity of time is changed. Moreover, we would like to formally specify mechanisms for applying sanctions upon detecting a violation that take, for example, the delay of fulfilling an obligation into account.

For verification purposes, it would be useful to have a real-time logic for reasoning about normative computational systems with real-time deadlines. Here metric temporal logic [16] seems to provide a particularly good starting point in which one can, for example express so-called time-bounded response formulas. These are of the form $\Box(p \rightarrow \Diamond_{\leq b} q)$, and informally express that it should always be the case that if p occurs, then q occurs within b time units. Formulas like these in the formal language of metric temporal logic also match closely with norms of the form (p, q, b) that we have used. We plan to investigate this relation in more detail.

9. REFERENCES

[1] R. Alur and T. A. Henzinger. Logics and models of real time: A survey. In *Real-Time: Theory in Practice*, volume 600 of *LNCS*, pages 74–106. Springer, 1992.

[2] L. Astefanoaei, F. de Boer, and M. Dastani. Strategic executions of choreographed timed normative multi-agent systems. In *Proceedings of the nineth international joint conference on autonomous agents and multiagent systems (AAMAS'10)*, pages 965–972. IFAAMAS, 2010.

[3] G. Boella, J. Broersen, and L. van der Torre. Reasoning about constitutive norms, counts-as conditionals, institutions, deadlines and violations. In T. D. Bui, T. V. Ho, and Q.-T. Ha, editors, *Proceedings of the 11th Pacific Rim International Conference on Multi-Agents (PRIMA'08)*, volume 5357 of *LNCS*, pages 86–97. Springer, 2008.

[4] J. Broersen. Issues in designing logical models for norm change. In G. Vouros, A. Artikis, K. Stathis, and J. Pitt, editors, *Proceedings of the First International Workshop on Organized Adaption in Multi-Agent Systems (OAMAS'08)*, volume 5368 of *LNCS*, pages 1–17, 2009.

[5] S. Cranefield. A rule language for modelling and monitoring social expectations in multi-agent systems. In *Coordination, Organizations, Institutions, and Norms in Multi-Agent Systems (ANIREM'05 and OOOP'05)*, volume 3913 of *LNCS*, pages 246–258, 2006.

[6] M. Dastani, N. Tinnemeier, and J.-J. Ch. Meyer. A programming language for normative multi-agent systems. In V. Dignum, editor, *Multi-Agent Systems: Semantics and Dynamics of Organizational Models*. IGI Global, 2009.

[7] F. Dignum and R. Kuiper. Specifying deadlines with dense time using deontic and temporal logic. *International Journal of Electronic Commerce*, 3(2):67–86, 1998.

[8] G. Governatori, J. Hulstijn, R. Riveret, and A. Rotolo. Characterising deadlines in temporal modal defeasible logic. In *Proceedings of the 20th Australian joint conference on Advances in artificial intelligence*, pages 486–496. Springer-Verlag, 2007.

[9] G. Governatori, A. Rotolo, and G. Sartor. Temporalised normative positions in defeasible logic. In *Procedings of the 10th International Conference on Artificial Intelligence and Law*, pages 25–34. ACM Press, 2005.

[10] T. Henzinger, Z. Manna, and A. Pnueli. Timed transition systems. In *Proceedings of the Real-Time: Theory in Practice, REX Workshop*, volume 600 of *LNCS*, pages 226 – 251. Springer, 1991.

[11] T. A. Henzinger, Z. Manna, and A. Pnueli. An interleaving model for real time. Technical Report STAN-CS-90-1329, Stanford University, 1990.

[12] A. Hoare. *Communicating Sequential Procresses*. Prentice-Hall, 1985.

[13] J. Hübner, O. Boissier, and R. Bordini. A normative programming language for multi-agent organisations. *Annals of Mathematics and Artificial Intelligence*, 62:27–53, 2011.

[14] J. F. Hübner, O. Boissier, and R. H. Bordini. From organisation specification to normative programming in multi-agent organisations. In J. Dix, J. A. Leite, G. Governatori, and W. Jamroga, editors, *Proceedings of the eleventh international workshop on computational logic in multi-agent systems (CLIMA'10)*, volume 6245 of *LNCS*, pages 117–134, 2010.

[15] K. Irwin, T. Yu, and W. H. Winsborough. On the modeling and analysis of obligations. In *Proceedings of the 13th ACM conference on Computer and communications security (CCS'06)*, pages 134–143. ACM, 2006.

[16] R. Koymans. Specifying real-time properties with metric temporal logic. *Real-Time Systems*, 2:255–299, 1990.

[17] M. B. van Riemsdijk, K. V. Hindriks, C. M. Jonker, and M. Sierhuis. Formalizing organizational constraints: A semantic approach. In *Proceedings of the nineth international joint conference on autonomous agents and multiagent systems (AAMAS'10)*, pages 823–830. IFAAMAS, 2010.