A Framework for Monitoring Agent-Based Normative Systems

Sanjay Modgil  
Noura Faci  
King’s College London  
Dept of Computer Science  
London, United Kingdom  
sanjay.modgil@kcl.ac.uk

Felipe Meneguzzi  
Nir Oren  
King’s College London  
Dept of Computer Science  
London, United Kingdom  
felipe.meneguzzi@kcl.ac.uk

Simon Miles  
Michael Luck  
King’s College London  
Dept of Computer Science  
London, United Kingdom  
simon.miles@kcl.ac.uk

ABSTRACT

The behaviours of autonomous agents may deviate from those deemed to be for the good of the societal systems of which they are a part. Norms have therefore been proposed as a means to regulate agent behaviours in open and dynamic systems, where these norms specify the obliged, permitted and prohibited behaviours of agents. Regulation can effectively be achieved through use of enforcement mechanisms that result in a net loss of utility for an agent in cases where the agent’s behaviour fails to comply with the norms. Recognition of compliance is thus crucial for achieving regulation. In this paper we propose a generic architecture for observation of agent behaviours, and recognition of these behaviours as constituting, or counting as, compliance or violation. The architecture deploys monitors that receive inputs from observers, and processes these inputs together with transition network representations of individual norms. In this way, monitors determine the fulfillment or violation status of norms. The paper also describes a proof of concept implementation and deployment of monitors in electronic contracting environments.

Categories and Subject Descriptors

D.2.10 [Software]: Software Engineering; I.2.11 [Artificial Intelligence]: Distributed Artificial Intelligence—multi-agent systems

Keywords

Monitoring, norms, electronic contracts

1. INTRODUCTION

Recent years have witnessed a growing interest in the use of norms to regulate and coordinate agent behaviours, and so achieve the overall objectives of multi-agent systems. Two approaches have been taken. In the regimentation approach [9], adopted for example by electronic institutions [3], agent behaviours are constrained to those specified by norms. Hence, agent autonomy is drastically curtailed, and such regimented systems are less flexible in that only appropriately specified agents can join. In contrast, the enforcement approach [1, 2, 6, 14, 18] allows for autonomous agents, and hence the possibility of violation of norms by agents. Enforcement mechanisms are thus required to motivate agent compliance.

Copyright © 2009, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

by threatening some loss of utility for agents in the case of violation. The enforcement approach thus requires that agent actions are monitored; that is, they must be observable and recognised as complying with or violating norms, in order that the enforcement mechanisms be appropriately applied.

In this paper we describe a generic framework for monitoring of agent behaviours in normative multi-agent systems. We motivate and describe how our approach builds on the overbearing approaches to monitoring (e.g., [10]), whereby messages exchanged among agents are observed, and behaviours are inferred from these messages. This contrasts with intrusive approaches [8, 15, 20], which assume that the mental states of agents are available for inspection, thus making the design of agent-based systems more complex, and relying heavily on the compliance of agents to communicate the required data.

Our monitoring architecture accounts for the fact that deployment of agents in normative systems requires some assurance that enforcement mechanisms, such as punishments or sanctions, will be employed only as and when appropriate. Any such assurance will partly rely on some measure of certainty that a norm is reported as violated if and only if it has in actuality been violated. Two features of our approach to monitoring ensure, to the degree that it is possible, that such assurance can be provided.

1. Observers of agent behaviours are explicitly entrusted by the system’s participating agents to accurately report on these behaviours.
2. Trusted observers can be any environmental artifact, that may not only observe for messages exchanged, but also more generally report on whether some state of interest holds or not.

This paper also describes how individual norms — obligations, prohibitions, and permissions — can be represented as Augmented Transition Networks (ATNs) [21] that are processed by monitor agents, together with observations relayed to the monitors by trusted observers, in order to determine the fulfillment and violation status of norms. The key features of the ATN representation and processing are as follows.

1. An abstract, general model of norms is assumed.
2. Representation of complex behaviours and states of interest enacted and brought about jointly by groups of agents.
3. Only behaviours specified by the norms are represented; hence, a given ATN can represent the same norm specified in any one of a number of multi-agent systems.
4. ATN representations of norms are independent of each other, allowing run time addition and removal of norms.
Our approach is thus generic and applicable to a range of dynamic open normative systems, including normative organisations developed by the kinds of dedicated languages described in [2], as well as electronic contracting frameworks [18] in which contract clauses specify norms that the contract parties are beholden to.

In summary, this paper makes the following contributions to research on monitoring of norm-governed agent behaviours:

1. We propose a trusted observer model with observations of agent messages and states of interest, to provide some measure of assurance that enforcement mechanisms are appropriately applied, so encouraging deployment of agents in normative systems.

2. Norms are individually represented as independent ATNs that provide for monitoring of: complex, jointly realised agent behaviours, and; states of interest brought about by such behaviours.

3. Together, the above provide a general framework that is applicable to monitoring of norms in a broad range of open, dynamic multi-agent systems.

This paper is organised as follows. Section 2 describes some general normative concepts that our approach to monitoring makes use of. Section 3 then motivates and describes an architectural overview of our approach. In Section 4, we describe how individual norms are represented as ATNs, and processed by monitor agents. Section 5 describes validation of our approach. We report on a proof of concept implementation of a monitoring agent, and its processing of ATN representations of norms encoded in an electronic contract specified by the CONTRACT project\(^1\). The implementation demonstrates monitoring of AgentSpeak(L) agents [19] whose interactions are governed by normative clauses specified in an aerospace contract. Finally, Section 6 concludes with a discussion of future and related work. In particular we discuss how the framework presented here extends and generalises a preliminary framework described in [4].

2. A GENERAL MODEL OF NORMS

In this section we review a recent general model of norms [18] that distinguishes some general normative concepts shared by some existing work on norms and normative systems [12, 5], and which we adopt in this paper. In [18], a norm \( N \) is modelled as a tuple:

\[
(NormType, NormActivation, NormCondition, NormExpiration, NormTarget)
\]

where:

- \( N \) is said to come into force, or is activated, if the conditions, or state of interest, described by NormActivation hold. It is when \( N \) is activated, that one must monitor to ensure that the goal, or state of interest, described by NormCondition:
  - must be brought about by \( N \)'s NormTarget in the case that \( N \)'s NormType is an obligation;
  - may be brought about by \( N \)'s NormTarget in the case that \( N \)'s NormType is permission; or
  - must not be brought about by \( N \)'s NormTarget in the case that \( N \)'s NormType is prohibition.

\( N \)'s NormExpiration denotes the state of interest under which the norm is no longer in force.

Such states of interest describe states of the world in which actions have been performed (e.g., messages sent) or certain properties hold (e.g., ‘the temperature is maintained above 23 degrees for at least 90% of the time’). Finally \( N \)'s NormExpiration denotes the state of interest under which the norm is no longer in force.

Note that in this paper we monitor the class of obligations whose violation can be determined with respect to some temporal condition holding. Without a temporal condition on the satisfaction of an obligation to realise or achieve a state of affairs (an achievement obligation), it would not be possible to determine a point in time at which a violation had occurred. Obligations to maintain certain states of affairs (maintenance obligations) often also explicitly reference temporal conditions that can be evaluated to determine violation (consider the above mentioned obligation to maintain the temperature for a certain period of time). However, some maintenance obligations do not (e.g., an obligation to always drive on the left), and these will be considered further in future work in Section 6.

**Example 1.** Consider the norm — NormGoods — that describes an obligation on the purchaser of goods \( G \) from a supplier \( S \), where the purchaser is an organisational entity consisting of two agents: the buyer \( B \) and the financial department \( F \):

- **NormType** = obligation
- **NormActivation** = \( B \) is notified by \( S \) that goods \( G \) are in stock
- **NormCondition** =
  1. \( B \) must cancel the order within 7 days of receipt of notification
  2. \( B \) must accept the order within 7 days of receipt of notification and \( F \) must deposit payment for \( G \) in \( S \)'s bank account, within 3 days of \( B \)'s acceptance.

- **NormExpiration** =
  1. \( B \) has canceled the order within 7 days of receipt of notification or
  2. \( B \) has accepted within 7 days of receipt of notification and \( S \) has received payment for \( G \) from \( F \) within 3 days of \( B \)'s acceptance or
  3. \( S \) is declared bankrupt
- **NormTarget** = \( B, F \)

Recall that **NormExpiration** denotes the state of interest under which a norm is no longer in force. The above example illustrates that a norm may be deemed to have expired for reasons other than that the norm’s conditions have been fulfilled. In principle, extra conditions of the type ‘\( S \) is declared bankrupt’ may be encoded as exceptions in the activation condition. However, NormGoods may then inappropriately remain in force; suppose that \( S \) goes bankrupt three days after \( B \) is notified by \( S \) that goods \( G \) are in stock. The norm will remain in force if ‘\( S \) is declared bankrupt’ is not encoded in the expiration condition.

Finally, note that a number of options obtain as to how one formally represents the states of interest described by a norm’s components. In the following section we discuss how these options are determined by issues such as ease of recognition and trust, and how the choice of option may determine the readiness with which agents are deployed in normative systems.

\(^1\)www.ist-contract.org
3. TRUSTED OBSERVERS AND THE MONITORING ARCHITECTURE

3.1 Motivating Trusted Observers

Enforcement mechanisms are required to motivate agent compliance with norms. Hence, agent actions must be monitored; that is, they must be observed and recognised as complying with or violating norms, in order that the enforcement mechanisms can be appropriately applied.

Monitoring of compliance requires determining whether the states of interest described by Norm Activation, Norm Condition and Norm Expiration hold. In the overhearing approach to monitoring [10], the agent behaviours that bring about states of interest are inferred from the messages exchanged. This suffices when states are inherently described as ones in which messages have been sent and received (e.g., Norm Activation in Example 1). However, it may not be possible, or indeed desirable, to describe and recognise such states exclusively on the basis of exchanged messages. Consider Norm Condition in Example 1, in which:

F is obliged to deposit payment for G in S’s bank account within 3 days of B’s acceptance.

Fulfilment of this obligation can be recognised by observing for F’s sending of a notification message to S, informing the latter that payment has been made. However, to motivate deployment of S in a normative system containing the norm Norm Goods (e.g., as a signatory to an electronic contract specifying Norm Goods as a contractual clause), S must be assured that enforcement mechanisms will be applied if F does not pay. If recognition of fulfilment is solely based on observing the above notification message, then this assurance equates with an assurance that the notification message is sent if and only if payment has in fact been made. Of course, any such assurance would be questionable given that, if F has not made the payment, F can avoid sanction by still sending the notification message, since this will suffice to indicate fulfilment of the obligation.

We thus propose that agents deployed in normative systems explicitly entrust particular observers to accurately relay observations to monitors, where these observations may be of messages exchanged or of properties that describe some state of interest. Thus, in the above example, agents F and S might explicitly entrust the bank itself to be an observer that reports to a monitoring agent that F has deposited the money. Unlike the case of the notification message sent by F, no gain accrues to the bank if the bank mis-reports.

Alternatively, agents F and S could also entrust an observer other than the bank, to accurately report on both a message sent from F to S notifying that payment has been made, and a message sent from the bank to S notifying the latter that the money has been deposited (although observation of messages sent from banks would raise information privacy issues that may prove difficult to resolve).

Finally, consider the case in which a supplier is obliged to place a purchased book on an internet site for downloading by a purchaser, within 14 days of receiving payment. In this case, the internet site can be ascribed trusted observer status, reporting to a monitor that the book has been placed.

3.2 The Monitoring Architecture

In this section we describe our architecture for monitoring the behaviours of agents deployed in normative systems. Trusted observers report to monitors on whether states of interests referenced

by norms’ components, do or do not hold. Monitors process these observations together with augmented transition network (ATN) [21] representations of norms. In Section 4 we describe in detail how these ATNs represent the states of interest.

Figure 1 provides an overview of the monitoring architecture, in which the agents are treated as black boxes and their internal state transitions are invisible to the monitors. A mapper maps norms to their ATN representations, for input to the monitor (this input is provided off-line). At run time, monitors subscribe to all observers entrusted with reporting on the states of interest identified by a norm’s components (Norm Activation, Norm Condition and Norm Expiration). The monitor can identify which observers to subscribe to, based on the ATN representation (as will be described in Section 4). Notice that there is nothing in the specification of the monitor that ties it to a particular normative system.

At run-time, observers notify monitors as to whether states of interest hold or not, by notifying monitors of: messages exchanged amongst agents, messages exchanged between agents and the environment, and predicates describing properties of the world (these properties may refer to actions having been performed). For example, in the use case implementation in Section 5, a state of interest in which an engine e has been repaired may be described in terms of a message sent from the repairer of the engine informing another party that the engine has been repaired, or an inferred predicate engine_repaired(e) from some theory describing the environment.

Notice that the observers are external to the normative system itself; their role is only to report on whether predicates hold or not, or on the sending and receipt of messages. They are not responsible for any kind of processing of this information. Thus, any environmental artifact can be assigned trusted observer status, including internet sites, human agents, banks, description logic reasoners, etc.
Monitors process observations together with the ATN representations of the norms, to determine when a norm is activated, fulfilled or violated, or has expired. Finally, the monitor informs manager agents of norms that have been violated, and of the agents responsible for violation. Manager agents then in turn impose sanctions on responsible agents.

Notice also that the choice of observers (and monitors) is application-specific, and agreed to by the agents whose behaviours are being observed, where such agreement constitutes a declaration of trust. The behaviours of observers (and monitors) may themselves be governed by normative clauses, and thus observed and monitored for deviation from their expected behaviour. This would reduce the potential for collusion (e.g., an agent dealing with eBay is more likely to trust a PayPal observer, even though PayPal is owned by eBay, if the behaviour of PayPal is itself normatively prescribed and sanctioned in case of violation).

4. REPRESENTATION AND PROCESSING OF NORMS FOR MONITORING

ATNs are essentially directed labelled graphs that were originally proposed for parsing complex natural languages. In general, transitioning an arc from one ATN node to another is based on some processing of that arc’s labels. A basic ATN representation of a norm \( N \) is a three node ATN \((S_1, S_2, S_3), A_1, A_2)\), where \( A_1 \) is the set of arcs connecting node \( S_1 \) to node \( S_2 \), and \( A_2 \) is the set of arcs connecting node \( S_2 \) to node \( S_3 \). Here, the transition from \( S_1 \) to \( S_2 \) corresponds to the activation of a norm, while the transition from \( S_2 \) to \( S_3 \) corresponds to the norm’s fulfillment or violation. The arcs are labelled so that, based on reports received from observers, the monitor matches the reports with the arc labels that describe the states of interest specified by \( N \)’s components, and then transitions the ATN from one node to the next across the matched arc. If the ATN remains at node \( S_1 \), then \( N \) is not activated. If the ATN is at node \( S_2 \), then \( N \) is activated, and in the case of an obligation, if it remains at \( S_2 \) after a given period of time identified by \( \text{NormCondition} \), then \( N \) is violated (the deadline is exceeded). If the ATN is at node \( S_3 \), then \( N \) is fulfilled if \( N \) is an obligation or permission, or violated if \( N \) is a prohibition (a prohibited state has been reached).

In what follows we describe in more detail the representation of norms as ATNs, and their processing by monitors.

4.1 Representing Norms as ATNs

A key requirement for norm representation and processing is to account for representation of complex behaviours and states of interest, possibly enacted and brought about jointly by multiple agents. For example, consider \( \text{NormGoods} \) (Example 1) in which the state in which the obligation is fulfilled is described by a disjunction of behaviours, where the second disjunction:

\[ B \text{ must accept the order within 7 days of receipt of notification and } F \text{ must deposit payment for } G \text{ in } S \text{'s bank account, within 3 days of } B \text{'s acceptance} \]

describes a conjunction of behaviours by two distinct agents \( B \) and \( F \), where these obliged behaviours must satisfy some temporal constraints. Recall also Section 3.1’s suggested representation for detection of \( F \) must deposit payment for \( G \) in \( S \)’s bank account, in terms of a notification message sent by \( F \) to \( S \) and a message from \( S \)’s bank, to \( S \), confirming receipt of the monies.

In order to be able to represent such complex descriptions of behaviours and states of interest, we assume representation of each norm component in disjunctive normal form. In Section 3 we described requirements for representing the states of interest described by each norm component, in terms of messages exchanged, and/or predicate descriptions. Such representations are obtained by a mapping from a disjunctive normal form expression of each norm component. These representations are then used to label arcs in the ATN representation of a norm.

**Definition 1.** [Mapping norm components to ATN labels]

For each norm component \( N \text{Comp} \in \{ \text{NormActivation}, \text{NormCondition}, \text{NormExpiration} \} \), let \( N \text{Comp} = \alpha_1 \lor \ldots \lor \alpha_n \), where for \( i = 1 \ldots n \), \( \alpha_i \) is a conjunction:

\[ \beta_1 \land \ldots \land \beta_m. \]

Then, for \( j = 1 \ldots m \):

\[ \text{map}(\beta_j) = \{ (\text{Ob}_{\beta_j}, \text{M}_{\beta_j}, T_{\beta_j}) \} \], where:

1. \( M_{\beta_j} \) is a message or predicate description
2. \( T_{\beta_j} \) is a temporal expression to be evaluated in conjunction with the processing of \( M_{\beta_j} \)
3. \( \text{Ob}_{\beta_j} \) is a unique identifier for the observer responsible for observing and reporting \( M_{\beta_j} \)

For any \( \alpha_i = \beta_1 \land \ldots \land \beta_m \) we let

\[ \text{map}_\text{conj}(\alpha_i) = \bigcup_{j=1}^m \text{map}(\beta_j) \]

Notice that a predicate description may be of form \( \text{happened}(\text{Act}) \) where \( \text{Act} \) is an action, or may describe the post-conditions of an action. The latter option may provide for more flexibility in terms of the actions executed to bring about the post-conditions describing the state of interest. While \( \text{Ob}_{\beta_j} \) uniquely identifies the observer entrusted to report the messages or the truth of predicates in \( M_{\beta_j} \), in practice, multiple observer identifiers may refer to the same observer. Notice that the observer identifiers enable a monitor to identify which observers to subscribe to when processing ATNs.

**Example 2.** For \( \text{NormActivation} \) in Example 1, we obtain \( \{ (\text{Ob}_B, \text{notify}, \text{send}(S, B, \text{in}_\text{stock}(G)), T_1^1), T_1 \} \) and for the first disjunct in \( \text{NormCondition} \):

\[ \{ (\text{Ob}_B, \text{cancel}, \text{send}(B, S, \text{cancel}(G)), T_2^2, T_2 \leq (T_1 + 7)) \} \]

and for the second disjunct in \( \text{NormCondition} \):

\[ \{ (\text{Ob}_B, \text{accept}, \text{send}(B, S, \text{accept}(G)), T_3^3, T_3 \leq (T_1 + 7)), \text{Ob}_2, \text{bank}, \text{deposit}(F, S, \text{Gcost}), T_4^4, T_4 \leq (T_3 + 3) \} \]

We now define representation of a norm as an ATN:

**Definition 2.** [Defining ATN representations of norms]

Let \( N = (\text{NormType}, \text{NormActivation}, \text{NormCondition}, \text{NormExpiration}, \text{NormTarget}) \), where:

- \( \text{NormActivation} = \delta_1 \lor \ldots \lor \delta_m \)
- \( \text{NormCondition} = \epsilon_1 \lor \ldots \lor \epsilon_n \)

The defined ATN is a three node labelled graph represented as a tuple \((\{S_1, S_2, S_3\}, \{A_1, A_2\}, \text{map}_\text{conj}()) \)

\[ \text{map}_\text{conj}() \quad \text{such that for } i = 1 \ldots m \text{, map}_\text{conj}() \text{ labels } (S_1, S_2), \text{ and } A_2 \text{ is a set of arcs } \{(S_2, S_3), \ldots \{(S_2, S_3)\} \text{ such that for } i = 1 \ldots m \text{, map}_\text{conj}() \}

Figure 2a illustrates an ATN representation of a generic norm \( N \), and Figure 2b shows the ATN representation of \( \text{NormGoods} \).

*We assume FIPA communication standards, and represent message descriptions as tuples consisting of the message itself and its time stamp*
4.2 Interpretation of ATNs by Monitors

This section describes how monitors process reports received from observers, together with ATN representations of norms, in order to determine the status of norms.

Consider a norm $N$ represented as the ATN $(\{S_1, S_2, S_3\}, A_1, A_2)$. If at least one arc in $A_1$ or $A_2$ is satisfied — by which we mean that observers report that the messages labelling the arc have been sent, and $\lor$ or the predicate descriptions hold, and the temporal expressions labelling the arc are evaluated by the monitor to be true — then the monitor transitions the arc to the successor node (where $S_2$ and $S_3$ are the successors of $S_1$ and $S_2$ respectively).

If an arc $\delta_i$ in $A_1$ is transitioned from $S_1$ to $S_2$, the monitor can communicate to a manager that $N$ is activated, which reasons for the activation are the messages and predicates labelling $\delta_i$. If an arc $\epsilon_i$ in $A_2$ is transitioned from $S_2$ to $S_3$, and $N$ is a permission or obligation, then the monitor can communicate to a manager that $N$ is fulfilled; if $N$ is a prohibition, then the monitor can communicate to a manager that $N$ is violated. In either case, the reasons for the fulfillment or violation — the messages and predicates labelling $\delta_i$ — can also be communicated to the manager.

Finally, NormExpiration represents conditions under which, if an ATN is in node $S_1$, the norm is not activated (irrespective of whether an arc in $A_1$ is satisfied), and if the ATN is in state $S_2$, then the issue of the norm’s fulfillment / violation no longer arises.

4.2.1 Time windows for detecting violation of obligations

We have thus far not considered the case in which an obligation is violated. We aim at representation and monitoring of obligations whose violation can be determined with respect to some temporal condition holding. For example, in order that one can determine whether an obligation to bring about some state (an achievement obligation) is violated, reference is required to some point of time before which the state must be brought about. Some main-tenance obligations also explicitly reference a time period during which some state of affairs must be maintained.

We thus define the notion of a time window, based on the temporal expressions labelling the arcs in $A_2$. Consider a modified version of Example 1 in which we simply have the obligation $\beta_1 = \‘B\text{ is obliged to cancel within }7\text{ days of receipt of notification}'$. This obligation can only be violated after 7 days after receipt of notification. 7 days is the time window for this obligation. Consider also a modified version of Example 1 in which $\beta_2 \wedge \beta_3 = \‘B\text{ is obliged to accept within }4\text{ days of notification and }F\text{ pays within }6\text{ days of notification}'$. This obligation can only be violated after 4 days after receipt of notification. Notice that if it is day 5, and $B$ has not accepted, then the obligation is violated. Now let us put these normative conditions together in a disjunct: $\beta_1 \lor (\beta_2 \wedge \beta_3)$. The time window in which the monitor must check for violation is now the maximum time window of all the disjuncts, i.e., 7 days. To illustrate, consider the following scenarios:

1. Suppose it is day 6 and neither disjunct is satisfied. The obligation is not violated since $B$ still has a day left in which to cancel.
2. Suppose it is day 8 and $B$ has accepted on day 5 after the notification and $F$ has paid on day 5 after the notification. The obligation is violated since neither disjunct is satisfied.
3. Suppose it is day 7 and $B$ has accepted on day 4 after the notification and $F$ has not yet paid. The obligation is not violated since it can still be fulfilled by $B$ by cancelling on day 7.

In general then, we assume a defined function time_window that returns a temporal expression (e.g. \‘7 days after receipt of notification\’) which, if evaluated to true, and if none of the NormCondition’s arcs (in $A_2$) are satisfied, then if the norm is an obligation, the norm is said to be violated.

**DEFINITION 3.** [Signatures of functions defining time windows]

Let $A_2 = \{ (S_2, S_3) \}, \ldots, (S_2, S_3)_n \}$. Then:

- $time_w_d : (S_2, S_3), \rightarrow \text{TW}_D \text{ where } \text{TW}_D \text{ is a temporal expression.}$
- $time_window : \{ time_w_d((S_2, S_3)_1), \ldots, time_w_d((S_2, S_3)_n) \} \rightarrow \text{TW} \text{ where } \text{TW} \text{ is a temporal expression}$

From the preceding example, one can see that an obvious way to define these functions is by letting $time_w_d$ return some minimal temporal value for temporal expressions in the conjuncts labelling a single arc, and $time_window$ returns the maximum amongst these values. In what follows, we will as an abuse of notation write $time_window(A_2)$ to denote the temporal expression returned by the function $time_window$ in the above definition.

4.2.2 Formally defining interpretation of ATNs

We now define the above concepts formally, in which we assume a monitor $Mon$ interpreting an ATN $\mathcal{X}_N = \{ (S_1, S_2, S_3) , A_1, A_2 \}$ for the norm $N = (\text{NormType, NormActivation, NormCondition, NormExpiration, NormTarget})$.

**DEFINITION 4.** [Satisfaction of arcs]

Let $a$ be an arc in $A_1$ or $A_2$, where $a$ is labelled by:

\{ $(\text{Ob}_{\beta_1}, M_{\beta_1}, T_{\beta_1}), \ldots, (\text{Ob}_{\beta_n}, M_{\beta_n}, T_{\beta_n})$ \}

We say that $a$ is satisfied, if for $i = 1 \ldots n$, $\text{Mon}$ receives $M_{\beta_i}$ from $\text{Ob}_{\beta_i}$, and $T_{\beta_i}$ evaluates to true.

\[This\, illustrates\, that\, one\, should\, additionally\, prohibit\, an\, agent\, from\, both\, accepting\, and\, cancelling.\]
DEFINITION 5. [Interpreting NormExpiration]
Let NormExpiration = γ₁ ∨ ... ∨ γₙ.

- We say that a monitor Mon evaluates NormExpiration as not holding if, for j = 1...n, γ_j does not not hold.
- We say that γ_j = (β₁ ∧ ... ∧ β_m) does not not hold if:
  - for at least one i, where map(β_i) = (Obᵦᵦ, Mᵦᵦ, Tᵦᵦ), either Mon does not receive Mᵦᵦ from Obᵦᵦ, or Tᵦᵦ does not evaluate to true.

DEFINITION 6. [Transitioning ATN to S2]
Let N be not activated, fulfilled, or violated (XN is in node S1). Suppose monitor Mon evaluates NormExpiration as not holding. Then, if at least one arc a in A1 is satisfied, XN transitions to S2 and N is said to be activated.

DEFINITION 7. [Transitioning ATN to S3]
Let N be activated (XN is in node S2). Suppose monitor Mon evaluates NormExpiration as not holding. If at least one arc a in A2 is satisfied, then XN transitions to S3, and:

- If N is an obligation or permission, then N is said to be fulfilled.
- If N is a prohibition then N is said to be violated.

DEFINITION 8. [Violation of obligation in S2]
Let the obligation N be activated (XN is in node S2). Suppose Mon evaluates NormExpiration as not holding. If time_window(A2) evaluates to true, and no arc in A2 is satisfied, then XN remains in node S2, and the obligation N is said to be violated.

We have thus far illustrated monitoring with respect to the obligation NormGoods in Example 1. In the next section we describe an implementation of a monitor, and its validation under a real-world scenario that involves agents governed by permissions, prohibitions and obligations in the aerospace domain.

5. IMPLEMENTATION AND VALIDATION

5.1 Monitor Implementation

This section describes an implementation of a monitor that receives messages from observers, and processes them so as to transition the ATN representations of the norms being monitored. At its core, our monitor contains a message store that is updated by received messages. When an arc is satisfied (see Definition 4) with respect to the contents of a message store, the monitor transitions the ATN. We consider two types of ATN: abstract and instantiated. ATNs in state S1 are said to be abstract because their arcs are labelled by expressions whose variables will be instantiated by concrete situations in which the norm comes into force (is activated). Hence, given an abstract ATN, when an arc a in A1 is satisfied, the resulting grounding of the variables in the Mᵦᵦ, labelling a is propagated to the variables in expressions labelling arcs in A2, thus creating an instantiated instance I of the abstract ATN, where I is then transitioned to S2 (corresponding to activation of the norm).

For example, the norm in Example 1 has a variable denoting a generic type of good that, once accepted, needs to be paid for. In this case, when the monitor receives a message denoting that a supplier Susan has the good galoshes in stock for buyer Bernard, the abstract ATN of Figure 2b is instantiated and its variables S, G and B are bound to Susan, galoshes and Bernard respectively. From this point on, only observer messages whose contents satisfy these variable groundings can cause the instantiated ATN to be transitioned to state S3.

Finally, when norms are fulfilled or violated, the monitor generates notifications to the manager to take appropriate action. We illustrate this operation in Figure 3, which shows the flow of messages from the observers to the monitor’s message queue, its processing and subsequent notifications to the manager.

Algorithm 1 Monitor control loop

<table>
<thead>
<tr>
<th>Require: Message queue Qₘₕₖ</th>
<th>Require: Message store Mₛ</th>
<th>Require: Set of abstract norm ATNs Xₐₖₖ</th>
<th>Require: Set of instantiated norm ATNs Xᵢₙₛₖ</th>
</tr>
</thead>
<tbody>
<tr>
<td>while Monitor is active do</td>
<td>while Qₘₕₖ is not empty do</td>
<td>Retrieve Msg from head of Qₘₕₖ</td>
<td>Add Msg to Mₛ (First, deal with messages)</td>
</tr>
<tr>
<td>for all Abstract norm ATN A in Xₐₖₖ do</td>
<td>for all Arcs α in A₁ do</td>
<td>if satisfies(Mₜₛ, arc label α) then</td>
<td>if satisfies(Mₜₛ, arc label α) then</td>
</tr>
<tr>
<td>Add A to Xᵢₙₛₖ</td>
<td>move I to state S2</td>
<td>create a norm ATN instance I of A</td>
<td>remove I from Xᵢₙₛₖ</td>
</tr>
<tr>
<td>end if</td>
<td>end if</td>
<td>add to I to Xᵢₙₛₖ</td>
<td>end if</td>
</tr>
<tr>
<td>end for</td>
<td>end for</td>
<td>move I to state S3</td>
<td>notify manager of violation</td>
</tr>
<tr>
<td>end if</td>
<td>end if</td>
<td>else if Norm I is a prohibition then</td>
<td>notify manager of violation</td>
</tr>
<tr>
<td>end if</td>
<td>end if</td>
<td>end if</td>
<td>end if</td>
</tr>
<tr>
<td>end while</td>
<td>end while</td>
<td>end if</td>
<td>end while</td>
</tr>
</tbody>
</table>

Figure 3: Overview of the monitor control loop.

This process is more precisely illustrated in Algorithm 1, which describes the control loop used in our monitor.

Its initial loop (lines 2 to 27) handles messages received from the observers by initially trying to satisfy outgoing arcs from S1 in the abstract ATNs (line 6). If successful for at least one arc (line 7), then an instantiated ATN is created, added to the set of instantiated ATNs and transitioned to S2 (lines 8 to 10). Satisfaction of out-
5.2 Use Case

In order to validate our approach to monitoring, we implemented and deployed a monitor in a prototype multi-agent system in which agents exchange messages that correspond to obliged, prohibited and permitted behaviours encoded in an electronic contract. Specifically, recent work on electronic representations and software tools for contracts [18] have highlighted a number of case studies [7]. A prototype for an aerospace logistics case study [16] implements aerospace agents — airline operators (AOs), engine manufacturers (EMs), and service sites (SSs) — whose behaviours are required to comply with (amongst others) norms governing the repair of engines and sourcing of parts for these repairs. In particular, it is commonplace for EMs, located at airports, to be under obligation to have operational engines available for the planes of a client AO. Furthermore, AOs may dictate permissions and prohibitions on the sourcing of parts for their engines. These norms are then inherited in contracts between EMs and service sites responsible for the actual servicing and repair of engines. For instance, in order for a given EM Boing to fulfill its obligations and provenance restrictions for a given AO Boing’s contract C with a service site Heathedge stipulates the following norms:

1. **Ob1**: Heathedge is obliged to repair engines for Boing within 7 days of receipt of an order for repair.
2. **Per1** and **Per2**: Heathedge is permitted to source parts for engines for Boing, from part manufacturers 1 and 2 (pm1 and pm2).
3. **Pro3**: Heathedge is prohibited from sourcing parts for engines for Boing, from part manufacturer 3 (pm3).

The prototype implementation deploys AgentSpeak(L) agents [19] representing each of the contract parties Boing and Heathedge, as well as the part manufacturers pm1, pm2 and pm3. We also deploy observers, a manager, and a monitor. The observers are responsible for relaying to the monitor, messages exchanged between Boing and Heathedge, and between Heathedge and the part manufacturers. In what follows we describe two scenarios in which messages are observed and relayed to a monitor who then processes these together with ATNs representing the above norms, and reports on their status. For both scenarios it is assumed that there is a prior message from Boing to Heathedge ordering repair of an engine. This message results in the ATNs for each of Ob1, Per1, Per2 and Pro3 being transitioned to their activation state S2.

**Scenario 1**:

1. Heathedge orders the part from pm1. This order message results in the ATN for Per1 transitioning to S3, then indicating fulfilment of the permission.
2. However, Heathedge is notified by pm1 that the part has been sent on the 4th day after the part order from Heathedge (because of unavoidable delays).
3. Because of the delay in receipt of the part, Heathedge cannot repair the engine repair within 7 days and so no message informing completion of repair is sent from Heathedge to Boing within the 7 day time window. Hence the ATN for Ob1 is in state S2 on day 8, and the monitor informs the manager that Heathedge has violated its obligation.

In Scenario 2, pm1 and pm2 inform Heathedge of delivery times that are not acceptable. Heathedge is then faced with a choice of violating its obligation, or ordering from the prohibited pm3 who can provide the part in an acceptable time. Hence, a given ATN chooses the latter option. Observation of the order message sent from Heathedge to pm3 results in transitioning the ATN for Pro3 to S3, and the monitor informs the manager of violation of this prohibition. However, Heathedge informs Boing of completion of repair on the sixth day after receipt of the repair order. This results in the ATN for Ob1 transitioning to S3. Hence, the obligation is reported by the monitor as fulfilled.

6. CONCLUSIONS

In this paper we have described a monitoring architecture in which trusted observers report to monitors on states of interest relevant to the activation, fulfillment, violation and expiration status of norms. This provides some measure of assurance that sanctions will be applied as and when appropriate, and thus enhances prospects for deployment of agents in normative systems. We have described how individual norms are represented as ATNs, that are processed by monitors together with observations relayed by observers, in order to determine the status of norms. The ATNs represent complex behaviours and states of interest, and only behaviours specified by the norms are represented. Hence, a given ATN can represent the same norm specified in any one of a number of multi-agent systems. Furthermore, ATN representations of norms are independent of each other, allowing run time addition and removal of norms to the monitored system. Taken together, these features enhance the generality of the framework, and to substantiate this claim, we have described a proof of concept implementation of a monitor agent monitoring agents whose behaviours are governed by normative clauses in an electronic contract.

The framework described in this paper extends a recent preliminary framework [4] in that it allows for monitoring of complex jointly realised behaviours, deploys trusted observers, and implements and validates processing ATN representations of norms. Other work related to ours includes monitoring of contracts in a Web Services context [11, 17], where the focus is on quality of service metrics rather than on the behaviours of agents. Other researchers adopt an overhearing approach to monitoring in organisational contexts [13, 10]. However, these works adopt overhearing in order to infer the mental states of the agents, where these states are domain-dependent and private to the agents. By contrast, we adopt overhearing of messages exchanged and predicates, for the different purpose of evaluating the status of norms.

We conclude with a discussion of three areas of future work to be pursued. As mentioned in Section 2, further work needs to address representation and monitoring of maintenance obligations that do not explicitly refer to temporal conditions. For example, consider an obligation to always drive on the left. In practice, such a norm
will be activated at a time $T_1$ when an individual starts to drive, and terminates at time $T_2$ when she ceases to drive. One might then specify an explicit temporal window: $\forall T, \ T_1 \leq T \leq T_2$. This would require extending the processing of an ATN to transition back and forth from $S_2$ to $S_3$. At each time point $T$, if she is driving on the left, the ATN transitions to the fulfillment state $S_3$, and then immediately transitions back to $S_2$, so that if at the next time point she is not driving on the left, then she can be detected as having violated the obligation. An alternative would be to create a new instantiation of the maintenance obligation at each time point $T$. In this view, the fulfillment of an instance of the norm at $T$ results in its expiration, and so a new instance is created at $T + 1$.

Providing explanations of norm violations is important if managers are to appropriately assign responsibility and apply sanctions. Explanations can also help to evolve the normative specification of a system in order to prevent future violations. Thus far, monitors provide limited explanation of violation and fulfillment, in terms of the labels of the arcs transitioned. Future work will investigate generation of more comprehensive explanations. This may require reference to representations of work-flow separate from that implicitly specified by norms. This information may in turn be gleaned from observers. For example, in the second scenario in the use case, access to all messages exchanged between Heathedge and the part manufacturers can be used to explain that in order to fulfill the repair obligation, the prohibition on sourcing parts from $pm3$ had to be violated, since $pm1$ and $pm2$’s delivery times would not have allowed the obligation to be fulfilled.

Finally, we note that the focus of this paper has been on corrective monitoring, whereby critical states are monitored for violation of norms. Predictive monitoring requires representation and recognition of danger states, which are associated with agent behaviours that suggest that a norm may be in danger of violation. Future work will address how such states may be identified empirically; for example by observing and analysing violation of norms at runtime and the intermediate states that are reached prior to violation. These intermediate states can then be explicitly (as extra nodes) in the ATN representation of norms, so that during future run-time executions, observation of messages indicating transition to these states may signal preemptive action to avoid violation.

Acknowledgements: The research described in this paper is partly supported by the European Commission Framework 6 funded project CONTRACT (INFSO-IST-034418). The opinions expressed herein are those of the named authors only and should not be taken as necessarily representative of the opinion of the European Commission or CONTRACT project partners.

7. REFERENCES