A Norm Enforcement Mechanism for a Time-Constrained Conditional Normative Framework

JAAMAS Track

Babatunde Opeoluwa Akinkunmi
Department of Computer Science,
University of Ibadan,
Ibadan, Oyo State, Nigeria
ope34648@yahoo.com

Florence Moyin Babalola
Department of Computer Studies,
The Polytechnic, Ibadan,
Ibadan, Oyo State, Nigeria
floxymbabs@gmail.com

ABSTRACT

This paper discusses the formalization of a norm enforcement mechanism for a regulative time constrained conditional normative framework. The norm representation captures both the condition and effect of a norm as situations and allows for rich temporal constraints between the times of the situations. As part of the enforcement process, a designated agent has an obligation within a time constraint to inform a liable agent of a reparative action they must take and the time constraint within which they must take it. That same agent must then monitor the compliance of the erring agent's obligation to carry out the reparative action, meting out sanctions in the case of violation.

KEYWORDS

Norm Enforcement, Conditional Norms, Temporal Constraints, Contrary to Duty Norms.

ACM Reference format:

B.O. Akinkunmi and Florence M. Babalola 2021. A Norm Enforcement Mechanism for a Time Constrained Conditional Normative Framework. In *Proceedings of International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2021) Online,May 3–7, 2021.* IFAAMAS, 3 pages.

1 Introduction

Time constrained conditional (tcc) norms have appeared in the normative reasoning literature (for example in the works of Governatori, Rotolo and Sartor[3] as well as Artikis and Sergot[2]). They may be reckoned to be an extension of the conditional normative structure such as those of Tossato et al[6] and Makinson and van der Torre[4] with a clear definition for the temporal constraint that exists between the times of norm's condition and effect, both of which are modeled as situations[5] .

This paper presents a norm monitoring and enforcement mechanism for tcc norms. The approach for norm enforcement presented is broadly through meting out sanctions and rewards, although one cannot rule out the enforcement of norms guiding the behaviour of enforcement agents through regimentation[1]. The enforcement takes places in three steps:

i. The detection of violation and conformance as well as the determination of the violating agent's liability; a violating agent is deemed to be liable with respect to violation only if they have failed to conform to all contrary to duty norms.

ii. The enforcement of a norm that obliges an enforcing agent to notify the liable agent of their obligation to carry out a reparative action within some time constraint of receiving the message. The enforcing agent's obligation to notify also has its time constraint.

iii. The enforcement of the obligation passed from the enforcing agent to the violating agent through meting out sanctions for any violation of that obligation.

The rest of the paper addresses the basic theory behind the enforcement mechanism and its logic programming implementation.

2 Theoretical Framework

The notion of a norm is defined around two concepts: that of a normative rule and that of a normative token that derives from it.

Definition 2.1

A norm or a normative rule is a five-tuple (nf, tc, p, r, nid) where:

- nf is normative fluent of the kind: obl(a) or pro(a) which refers to either an obligation to carry out or a prohibition from carrying out an action of type *a*.
- tc is a time constraint.
- p is a property of situations
- r is a role that an agent may in s situation, and
- nid is the identity of the norm.

A normative token can be derived from a normative rule as shown in the following definition:

Definition 2.2

A normative token, a 5-tuple (ag, nf, s, tc, nid) exists if and only if: ag is an agent, s is a situation and all of nf, tc and nid are as defined in Definition 2.1 and there exists a normative rule (nf, tc, p,

Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, Online. © 2021 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

r, nid) such that: s meets the property p, i.e. p(s) is true and ag plays the role r in the situation s.

2.1 Norm Monitoring: Violation, Conformance and Liability

An agent is deemed to have carried out the action of violating an obligatory norm nid (with respect to a situation s) over a situation s_1 if the normative token (ag, obl(a), s, tc, nid) exists and ag has failed to carry out any action of type a within the maximal interval that satisfy the constraint tc with the time of s. the time of the situation s_1 over which the violation is deemed to have taken place is the maximal interval that satisfy the constraint tc with the time of s. The action is denoted by: Holds(occurring(do(ag, viol(obl(nid, s))), s_1)

An agent is deemed to have carried out the action of violating a prohibitive norm *nid* (with respect to a situation s) over a situation s_1 if the normative token (ag, pro(a), s, tc, nid) exists and ag has carried out the action type a over the situation s_1 and the time of the situation s_1 satisfies the time constraint tc prescribed in *nid* with the time of s. The fact that such an action holds over a situation is rendered as:

Holds(occurring(do(ag, viol(pro(nid, s))), s_1)

Similar actions are defined for a conformance actions, which is denoted by: Holds(occurring(do(ag, conf(pro(nid, s))), s_1) and Holds(occurring(do(ag, conf(obl(nid, s))), s_1).

A violation action committed by an agent is liable if and only if either the violated norm does not have any contrary-to-duty norm or the agent has violated at least one the violated norm's contrary-to-duty norms. The notation for expressing an action of liable violation is: Holds(occurring(do(ag, liable-viol(pro(nid, s))), s_1)

It is important to note that one instance of violation of any norm by a specific agent is distinguished from another by the particular situation that warranted the need for the norm's application.

2.2 Norm Enforcement

The enforcement mechanism is encoded in three norm-guided phases: The kinds of norms involved for each phase are:

- *Violation-notification norm*, which is an obligatory norm that encodes the responsibility of an enforcing agent to notify a violating agent of the need to carry out a reparative action within a certain time constraint. Each norm has its violation notification norm.
- *Notification response norm*, which is an obligatory norm that encodes the responsibility of a violating agent to carry out the reparative action prescribed for it in a violation notification norm within the time constraint prescribed.
- *Notification response enforcement norm*, which is an obligatory norm that encodes the responsibility of an enforcing agent to sanction a violating agent that also violates a notification response norm.

Thus, a violating agent will be sanctioned only when they fail to carry out a reparative action within the prescribed timeframe given by the enforcer in the notification response norm.

3 Logic Programming Implementation

The system described in this paper was implemented in a logic programming environment. Each norm is encoded as a logic programming rule of the form:

Normpos(ag, nf, s, tc, nid) *if*
$P(s) \wedge R(ag, s)$.

Thus given the pair (ag, s) such that ag is an agent and s is a situation that satisfies the property P and ag plays the role R in situation s then, the following fact is derived:

Normpos(ag, nf, s, tc, nid).

The above captures the notion of a normative token.

Individual norms from the domain of student activity such as: *the need for a student to register within the first 30 days of any semester*, were captured in this kind of form.

There are also logic programming rules for inferring violation, conformance and liable violation actions on the part of an agent. For example, the action of conforming with an obligation can be inferred from the following logic programming rule:

Holds(occurring(do(ag, conf(nid, s))), s_1) if
∃ act-type, tc,
Normpos(ag, obl(act-type), s, tc, nid) ∧
Holds(occurring(do(ag, act-type)), s_1) ∧
SatisfyCons(time(s_1), time(s), tc).

Finally, the same logic programming rule for representing domain norms also adequately captures the three kinds of norm enforcement norms.

4 Comparison to Other Formalisms

A somewhat simplified form of norm representation is what we refer to as *timed norms*. These norms prescribe an agent's obligation or prohibitions within specific time intervals[9, 10].

Notable differences exist between the notation developed here for tcc norms and other tcc norm notations in the literature: for example, violation norm inference by Artikis and Sergot[2] takes cognizance of the agent and their role, rather than a specific application of a norm. Again, the domain of Governatori et al[3] deals with obligations that persist in spite of violation by agents. Besides, our formalism has a clear representation for obligations or prohibitions that are prior to its activating conditions unlike the formalism of Oren *etal*[7] and Panagiotidi et al[8] . Examples of such norms are: *A robocar ought to have notified a garage at least 24 hours before it arrives for servicing* and *A person taking a fasting blood sugar test ought to have observed a 12 hour fast prior to the test.*

ACKNOWLEDGMENTS

The authors are grateful to the referees of the JAAMAS paper for their helpful comments.

REFERENCES

[1] Natasha Alechina, M. Dastani and B. Logan 2018 Norm Specification and Verification in Multi-agent Systems. *Journal of Applied Logics – IFCoLog Journal of Logics and their Application*, 5, 2, (April 2018), 457 -490.

[2] A. Artikis and M. Sergot, 2010. Executable Specification of Open Multi-Agent Systems, *Logic Journal of the IGPL*, 18, 1 (Jan, 2010), 31-65. DOI: https://doi.org/10.1093/jigpal/jzp071

[3] G. Governatori, A. Rotolo and G. Sartor 2005. Temporalised Normative Positions in Defeasible Logic . In *Proceedings of the 10th. International Conference on AI and Law* ACM Press, New York, NY, USA. 25-34. DOI: https://doi.org/10.1145/1165485.1165490

[4] David C. Makinson and L van der Torre. 2003. What is Input Output Logic, *Trends In Logic*, 17, 163-174.

[5] Lenhart K Schubert, 2000. The Situation We Talk About. In *Logic Based Artificial Intelligence*, Springer International Series in Engineering and Computer Science, 407-439.

[6] Silvano C. Tossatto et al 2012, Abstract Normative Systems: Semantics and Proof Theory, In *Proceedings of the 13th International Conference on the Principles of Knowledge Representation and Reasoning*, AAAI Press, 358 -368.

[7] Nir Oren et al 2009. Towards a formalization of Electronic Contracting Environment, In *Proceedings of the Workshop on Coordination, Organizations, Institution and Norms in Agent Systems*, Springer, Berlin, 156-171.

[8] Sofia Panagiotidi, Juan C. Nieves and Javier Vazquez-Salceda. 2009. A Framework to model Norm Dynamics in Answer Set Programming, In *Proceedings of the Second Multi-Agent Logics, Languages and Organizations Federated Workshops*, Turing, Italy.

[9] Fariba Sadri, K Strathis and Francesca Toni. 2006. Normative KGP Agents. *Computational &Mathematical Organization Theory*, 12, 2-3, 101 -126.

[10] Tiberiu Stratulat, Francoise Clerin-Debart and Patrice Enjalbert 2001, In Proceedings of the *8th International Symposium on Temporal Representation and Reasoning, (TIME, 01)*, Turing Italy, 41-47.