A Practical Resource-Constrained Norm Monitor (Extended Abstract)

Natalia Criado
King's College London
United Kingdom
natalia.criado@kcl.ac.uk

ABSTRACT

Norm monitoring often assumes that monitors have unlimited resources to observe the environment and the actions performed by agents. In this paper, we relax this assumption and propose a practical resource-constrained norm monitor capable of selecting the resources to be deployed based on their cost and value to norm monitoring.

1. INTRODUCTION

Norms allow system designers to specify the desired behaviour of a multi-agent system [6]. In many domains, regimenting these norms (i.e., making the violation of norms impossible) is too rigid and undesirable. In these domains, it is usually more adequate to employ norm enforcement mechanisms that persuade agents to comply with norms by imposing sanctions and rewards [15]. A key element of such norm enforcement mechanisms is norm monitoring; i.e., the process by which agent's actions are observed and checked against the norms. Norm monitoring requires the deployment of different types of resources that control the execution of some actions and also allow monitors to observe some properties of the environment. These resources can be expensive, and there is usually a limited budget that can be spent on monitoring norm compliance.

In this paper we propose a practical resource-constrained norm monitor capable of selecting the resources to be monitored based on their cost and value to norm monitoring.

2. PRELIMINARIES

$\mathcal{L}$ is a propositional language containing propositional symbols, the logical connective $\neg$, and the true (false) proposition $\top$ ($\bot$). We will relate our formulae via logical entailment $\vdash$ ($\nvdash$). The set of atomic formulas of $\mathcal{L}$ is built of a finite set of propositional symbols that characterise the properties of the world relevant to norm monitoring. Some of these properties are static and not altered by action execution, whereas other properties are dynamic and changed due to agent actions. Specifically, we represent static properties as a set[1] of atomic formulas of $\mathcal{L}$, denoted by g.

[1]In this paper sets are to be interpreted as the conjunction of their elements.

Appears in: *Proc. of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

2.1 Action Definition

In line with the existing literature [3], actions are represented as a tuple $\langle pre, post \rangle$ containing preconditions and postconditions, where these conditions are expressed as a set of literals of $\mathcal{L}$. Given an action a, we denote by $pre(a)$ and $post(a)$ the action precondition and postcondition.

2.2 Resource Definition

A resource is a set of coupled controls and sensors that have a unique cost. Controls allow agents to execute actions, whereas sensors allow some properties of the world to be observed. The cost of a resource can represent the economic cost of installing and maintaining it, the temporal cost associated with processing the sensor and control data obtained through the resource, etc. A resource is defined as a tuple $\langle control, sensor, cost \rangle$, where:

- *control* is the set of actions whose execution is controlled by the resource and that can be observed if the resource is deployed;
- *sensor* is the set of properties of the world whose truth value is observed when the resource is deployed;
- $cost \in \mathbb{R}_{\geq 0}$ is the cost of the resource.

Given a resource r, we define $control(r)$, $sensor(r)$ and $cost(r)$ as the actions and properties that can be observed through the resource and the cost of the resource. Given a resource r, we define the set of discernible literals, denoted by $dis(r)$ as:

$$\left(\bigcup_{\forall p \in sensor(r)} \{p, \neg p\} \right) \bigcup \left(\bigcup_{\forall a \in control(r)} pre(a) \cup post(a) \right)$$

We also extend these definitions to sets of resources.

2.3 Norm Definition

We consider *norms* as formal statements that define patterns of behaviour by means of *deontic modalities* (i.e., *obligations* and *prohibitions*) [9, 13, 15, 18]. In consonance with the related literature [1, 14, 2], we consider a "*closed legal system*", where everything is permitted by default, and obligation and prohibition norms define exceptions to this default rule. More formally, a *norm* is defined as a tuple $\langle deon, cond, activ, expir \rangle$, where:

- $deon \in \{\mathcal{O}, \mathcal{F}\}$ is the deontic modality, determining if the norm is an obligation ($\mathcal{O}$) or prohibition ($\mathcal{F}$);
- *cond* is the norm condition, i.e., the action whose execution is regulated by the norm;
- *activ* is a set of literals of $\mathcal{L}$ that represents the activation condition, i.e., the circumstances in which the norm becomes relevant;

- *expir* is a set of literals of $\mathcal{L}$ that represents the expiration condition, i.e., the circumstances in which the norm is no longer relevant.

Given a norm n, we define $cond(n)$, $activ(n)$ and $expir(n)$ as the norm condition, and the activation and expiration conditions, respectively.

3. NORM MONITOR

Definition 1. *A norm monitor is defined as a tuple $\langle A, R, N, b\rangle$ where: A is the set of actions that can be executed by agents; R is a set of resources that can be deployed; N is the set of norms that the monitor needs to check; and $b \in \mathbb{R}_{\geq 0}$ represents the budget of the norm monitor.*

The norm monitor has to select a set of resources satisfying the budget condition such that the detection of norm violations and fulfilments is maximal. This is an optimization problem such that:

$$\max_{S \subseteq R} v(S) \text{ subject to } cost(S) \leq b$$

where v is a function determining the value of a set of resources to norm monitoring.

3.1 Resource Value

We propose to calculate the value of a set of resources as an estimation of the number of norms whose compliance can be checked through these resources; i.e., the resources *coverage* of norm compliance. However, not all norms have the same probability of becoming relevant (e.g., norms whose activation condition is an empty set of literals are always activated), thus we weight each norm with its probability:

$$v(S) = \sum_{n \in N} pr(n) \times cover(S, n)$$

where $pr(n)$ is the *probability* of a norm and $cover(S, n)$ is the *coverage* of a norm provided by a set of resources.

We define the *probability* of a norm becoming relevant considering the probability of its activation being true, the probability in which the action in the norm condition can be executed and the probability of its expiration being false. For simplicity, we assume these three events are independent and we define the probability of a norm as follows:

$$\prod_{l \in activ(n)} pr(l) \times \prod_{l \in pre(cond(n))} pr(l) \times \max_{l \in expir(n)} pr(\neg l)$$

note that a set of literals is true when all their literals are true (i.e., we defined the probability of a set of literals as the product among the probability of each literal in the set) and it is false when just one of these literals is false (i.e., we defined this probability as the maximum among the probability of the negation of each literal in the set). The probability of a literal is determined by the chances it is or becomes true (i.e., the number of actions having as postcondition the literal):

$$pr(l) = \begin{cases} 1 & g \vdash l \\ 0 & g \vdash \neg l \\ \frac{|\{a | a \in A \wedge l \in post(a)\}|}{|A|} & \text{otherwise} \end{cases}$$

To check compliance with a norm, the monitor needs to be able to observe the execution of the norm condition and the truth value of the activation and expiration condition. Given a norm $n \in N$, we define its *coverage* by a set of resources S as the ratio of elements (i.e., action and literals) needed to check compliance that can be observed through this set of resources:

$$\frac{|cond(n) \cap control(S)| + |activ(n) \cap dis(S)| + |expir(n) \cap dis(S)|}{|cond(n)| + |activ(n)| + |expir(n)|}$$

The resource value function presents an interesting property that allows us to compute efficient suboptimal solutions with a given guarantee. In particular, this function is submodular; i.e., it exhibits a diminishing property: selecting a resource when few resources have been selected has more value (provides more information to check norm compliance) than selecting it after more resources have been selected.

3.2 CEF Algorithm

To select resources the monitor uses the CEF algorithm [12], which is a greedy algorithm for submodular function maximization. In particular, this algorithm performs: (i) a greedy search that uses the value to rank resources ignoring costs (i.e., the resource with the highest value is selected and added to the solution, this process is followed until it is not possible to add more resources without exceeding the budget); and (ii) a greedy search that uses the ratio value by cost to rank resources (i.e., the resource with the highest ratio value by cost is selected and added to the solution, this process is followed until it is not possible to add more resources without exceeding the budget). Then, the CEF algorithm returns the solution with the highest value. It has been demonstrated that this algorithm achieves an approximation guarantee of $\frac{1}{2}(1 - \frac{1}{e})$, whereas the empirical results show that this bound is much tighter.

4. DISCUSSION

The existing literature on norm enforcement has proposed several methods for monitoring norm compliance and applying sanctions.

First proposals on norm enforcement considered closed and relatively small multi-agent systems (MAS) and, thus, they proposed centralised architectures to enforce norm compliance [10, 5]. Later the interest switched from closed MAS to open MAS and there was a need for more efficient and scalable distributed architectures to norm enforcement [16, 15, 19, 11]. All of these proposals assumed that monitors have complete information about the actions performed by agents and the environment. In that case, checking norm compliance is easy as monitors have complete information on the triggering of norm violations (e.g., when forbidden actions are performed) and fulfilments (e.g., when obligatory actions are performed).

Exceptions to these complete monitoring approaches are three very recent proposals [4, 2, 7, 8] that address partial information about the environment and the actions performed by agents. In particular, they propose methods to overcome partial information by adding more monitors [4], limiting the norms to what can be observed [2], selecting agents under surveillance [7], and reconstructing unobserved actions [8]. Although these proposals allow the application of normative MAS into real world problems characterised by limited information, they neglect the cost of observing agent actions and the environment. Our paper presents the first resource-constrained norm monitor that is able to consider observation costs.

REFERENCES

[1] N. Alechina, M. Dastani, and B. Logan. Reasoning about normative update. In *Proc. of IJCAI*, pages 20–26, 2013.

[2] N. Alechina, M. Dastani, and B. Logan. Norm approximation for imperfect monitors. In *Proc. of AAMAS*, pages 117–124, 2014.

[3] C. Boutilier and R. I. Brafman. Partial-order planning with concurrent interacting actions. *Journal of Artificial Intelligence Research*, 14(1):105–136, 2001.

[4] N. Bulling, M. Dastani, and M. Knobbout. Monitoring norm violations in multi-agent systems. In *Proc. of AAMAS*, pages 491–498, 2013.

[5] H. L. Cardoso and E. C. Oliveira. Institutional reality and norms: Specifying and monitoring agent organizations. *International Journal of Cooperative Information Systems*, 16(1):67–95, 2007.

[6] N. Criado, E. Argente, and V. Botti. Open issues for normative multi-agent systems. *AI communications*, 24(3):233–264, 2011.

[7] N. Criado and J. M. Such. Selective norm monitoring. In *Proc. of IJCAI*, pages 208–214, 2016.

[8] N. Criado and J. M. Such. Norm monitoring under partial action observability. *IEEE Transactions on Cybernetics*, 47(2):270–282, 2017.

[9] F. Dignum. Autonomous agents with norms. *Artificial Intelligence and Law*, 7(1):69–79, 1999.

[10] M. Esteva, B. Rosell, J. A. Rodríguez-Aguilar, and J. L. Arcos. AMELI: an agent-based middleware for electronic institutions. In *Proc. of AAMAS*, pages 236–243, 2004.

[11] D. Gaertner, A. Garcia-Camino, P. Noriega, J.-A. Rodriguez-Aguilar, and W. Vasconcelos. Distributed norm management in regulated multiagent systems. In *Proc. of AAMAS*, pages 624–631, 2007.

[12] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *Proc. of SIGKDD*, pages 420–429, 2007.

[13] F. López y López, M. Luck, and M. d'Inverno. A normative framework for agent-based systems. *Computational & Mathematical Organization Theory*, 12(2-3):227–250, 2006.

[14] E. Lorini. On the logical foundations of moral agency. In *Deontic Logic in Computer Science*, pages 108–122. Springer, 2012.

[15] F. Meneguzzi, S. Modgil, N. Oren, S. Miles, M. Luck, and N. Faci. Applying electronic contracting to the aerospace aftercare domain. *Engineering Applications of Artificial Intelligence*, 25(7):1471–1487, 2012.

[16] N. Minsky and V. Ungureanu. Law-governed interaction: a coordination and control mechanism for heterogeneous distributed systems. *ACM Transactions on Software Engineering and Methodology*, 9(3):273–305, 2000.

[17] M. P. Singh. Norms as a basis for governing sociotechnical systems. *ACM Transactions on Intelligent Systems and Technology*, 5(1):21, 2013.

[18] W. Vasconcelos, M. J. Kollingbaum, and T. J. Norman. Resolving conflict and inconsistency in norm-regulated virtual organizations. In *Proc. of AAMAS*, pages 632–639. ACM, 2007.

[19] W. W. Vasconcelos, A. García-Camino, D. Gaertner, J. A. Rodríguez-Aguilar, and P. Noriega. Distributed norm management for multi-agent systems. *Expert Systems with Applications*, 39(5):5990–5999, 2012.