ECAI 2010
H. Coelho et al. (Eds.)
IOS Press, 2010

doi:10.3233/978-1-60750-606-5-1027

Reasoning about Norm Compliance with Rational Agents

Moser Silva Fagundes and **Holger Billhardt** and **Sascha Ossowski** [1]

Abstract. This paper presents a model for rational self-interested agents, which takes into account the possibility of violating norms. The transgressions take place when the expected rewards obtained with the defection from the norms surpass the expected rewards obtained by being norm-compliant. To develop such model, we employ Markov Decision Processes (MDPs). Our approach consists of representing the reactions for norm violations within the MDPs in such a way that the agent is able to reason about how those violations affect her expected utilities and future options.

1 INTRODUCTION

In Regulated Multiagent Systems (RMAS) a set of normative structures is set up to instil some desired global behaviour. Although autonomous agents may be persuaded to assume a particular behaviour, they are supposed to be free to accept or refuse to comply with a set of norms, and handle the consequences of their choices. An agent needs to be able to choose dynamically which norms to obey so as to successfully fulfil her purpose. Thus, to achieve an *adaptive behaviour*, the normative structures cannot be *hard-wired* into the agents [2].

A rational self-interested agent considers to violate a set of norms in those cases where the expected rewards obtained by the violation surpass the expected rewards obtained by being norm-compliant. By this we mean she estimates the earnings brought about by the violation and the losses caused by the reactions for the defections from the norms. Finally, based on such estimation, that agent is able to set a course of actions (possibly *non-compliant* with the norms) in order to maximize her expected rewards.

This paper presents a model for rational self-interested agents capable of adapting their policy to the norms in a self-interested way. During the adaptive process, the agent takes into account the possibility of disobeying (some of) the norms governing the RMAS. Our approach consists of updating the agents' knowledge in order to express the effects of the reactions taken against the transgressions. Consequently, she is able to reason about the outcomes of the violations in that particular RMAS, and then, adapt her behaviour by finding an adequate policy. To develop the agent architecture, we use Markov Decision Processes (MDPs).

2 REASONING ABOUT NORM VIOLATIONS

It is assumed that norms can be violated – *enforcement* instead of *regimentation* [4]. If a norm is violated, then a *reaction* takes place. To represent a norm, we propose the following form:

norm(*deontic value*, *agent*, *action*, *state*, *reaction*)

where:

- *deontic value* $\in$ {*prohibition, obligation*};
- *agent* $\in \Gamma$, and Γ is the set of agents participating in the RMAS;
- *action* $\in A$, and A is the *agent*'s action set;
- *state* $\in S$, and S is the *agent*'s state space;
- *reaction* has the form: reaction($outcome_{A(\cdot)}$, $outcome_T$), where:
  - $outcome_{A(\cdot)}$ contains the changes to be done in the agent's capability function: a table whose rows have the form ($state_i$, *action*, $\{0,1\}$), where 1 means the *action* is admissible in $state_i$;
  - $outcome_T$ consists of a table that stores the adjustments to be done in the probabilities of the agent's state-transition model; the rows have the form ($state_i$, *action*, $state_j$, $[0 \ldots 1]$), which indicates the probability of executing an *action* at $state_i$ and ending at $state_j$;

Figure 1 illustrates the agent's internal architecture. The components are depicted as grey boxes, while data is depicted as white rounded boxes. The set of *Norms* is the only external dataset. All other datasets, including the *Original MDP* [2], belong to the agent.

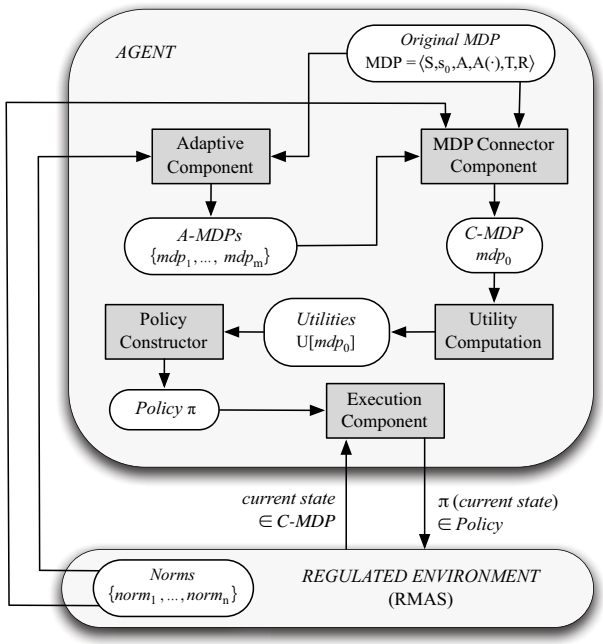

Figure 1. Overview of the agent's internal architecture.

[1] Centre for Intelligent Information Technologies (CETINIA), University Rey Juan Carlos, Madrid – Spain, email: moser.fagundes@urjc.es

[2] The initial knowledge of the agent is specified without taking into account any particular set of norms and reactions. This initial knowledge is specified as a Markov Decision Process, named *Original MDP*.

2.1 Adaptive Component

In this component, the agent makes the *decision* with respect to *what norms she will consider for violation*. Considering a norm for violation does not mean the agent defects from that norm, it means the agent will reason about it.

Once taken that *decision*, the *Adaptive Component* creates the set $\{mdp_1, \ldots mdp_m\}$, referred to as *A-MDPs*, where mdp_i, $1 \leq i \leq m \leq n$, is initially a replica of the *Original MDP*. Each *A-MDP* is created for representing the reactions associated with a particular norm. However, some norms might not have a corresponding *A-MDP* since the agent is able to consider just a subset of them. The adaptation of each *A-MDP* is done in two stages:

1. *Adaptation to represent reactions*. Represents within an *A-MDP* the effects of the reactions for violating a given norm. According to our normative structure, the reactions can affect the agents' capability function and state-transition model.
2. *Adaptation for norm-compliance*. Intends to prevent the violation of those norms *not* considered for violation. It is done by making shortenings in the *A-MDP*: a prohibited action a in a state s is prevented of being executed by removing it from the agent's capabilities, and an obliged action a, which must be executed in the state s, has its performance assured by suppressing all actions related to s, except the one representing the action a.

2.2 MDP Connector Component

Differently from the *Adaptive Component*, whose purpose consists of representing how the world would be if the agent violates a norm, the *MDP Connector Component* focuses on the construction of a single MDP, named *C-MDP*, by connecting the *Original MDP* with the *A-MDPs*. These *connections* are done by changing the outcomes of the violating actions. Instead of arriving exclusively at states of the current MDP, the execution of a violating action may lead to states of the *A-MDPs*.

Figure 2 illustrates how the connections between the MDPs are done. In this example, $norm_i$ indicates that the action a is *prohibited* for the *agent* in the state s_0. To model the chances of being caught and suffering the reactions, we replicate the transitions for the action a starting at the state s_0. But in place of arriving exclusively at states of the *Original MDP*, the new transitions arrive at their *analogous*[3] states $\{s_1, \ldots, s_k\}$ in the *A-MDP*($norm_i$). The probabilities of these new transitions are multiplied by P_i – the probability of the violation being detected. However, there is a chance of going undetected. To model this chance, we multiply by $(1–P_i)$ the state-transition probabilities arriving at states $\{s_1, \ldots, s_k\}$ of the *Original MDP*.

2.3 Final Remarks

The *Utility Component* computes the expected utility values for the states, while the *Policy Constructor* finds a policy based on those utilities. These two components address the question regarding *what norms are worthy of breaking*. A wide range of algorithms for implementing these components is available in the AI literature.

The probability of the violation being detected expresses the degree of impunity in the RMAS. It is an important information to be taken into account in the reasoning since a high degree of impunity is an incentive for the agent to assume a deviant behaviour.

[3] The *Original MDP* and the *A-MDPs* have identical state spaces, thus, any state of these tuples has exactly one *analogous* state in the other tuples.

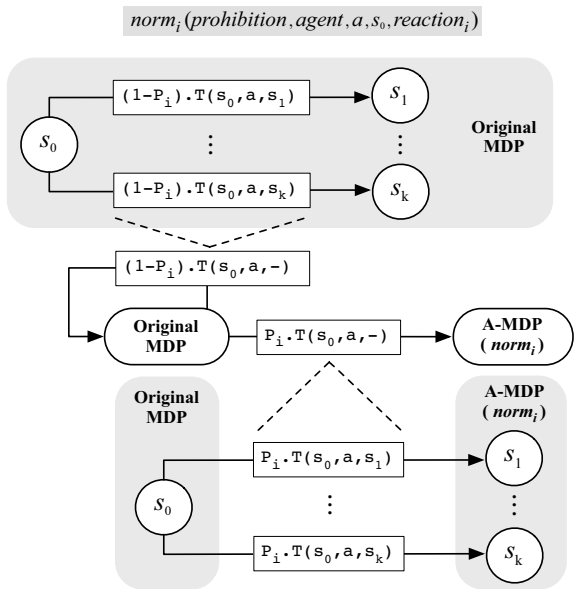

Figure 2. Rearrangement of the state-transitions T(s_0,a,–) of the *Original MDP* in order to establish the connection with the MDP($norm_i$). The value P_i stands for the the detection probability for the violation of the $norm_i$.

3 CONCLUSION

In this paper, we present an architecture based on MDPs for rational self-interested agents capable of estimating the benefits and losses for breaking a set of norms. Considering this estimation, the agent decides whether to comply or not with the norms. We do not explicitly generate normative goals [3, 1] despite the fact that they are implicitly represented within the norms. No preference values are associated with our normative structures. Instead, the impact of the norms on the agent is observed in her expected utilities and policy. Since we use MDPs, the planning costs and the plan benefits (expected utilities) are taken into account in the decision of whether or not comply with a norm. We do not represent explicitly the actions of the other agents, however, we consider the outcomes of the reactions, what for instance, correspond to actions taken by other entities in response to norm violations.

4 ACKNOWLEDGEMENTS

This work is supported by the Spanish Ministry of Science and Innovation through the projects "AT" (CSD2007-0022, CONSOLIDER-INGENIO 2010) and "OVAMAH" (TIN2009-13839-C03-02).

REFERENCES

[1] Jan Broersen, Mehdi Dastani, Joris Hulstijn, and Leendert van der Torre, 'Goal Generation in the BOID Architecture', *Cognitive Science Quarterly Journal*, **2**(3-4), 428–447, (2002).
[2] Cristiano Castelfranchi, Frank Dignum, Catholijn M. Jonker, and Jan Treur, 'Deliberative Normative Agents: Principles and Architecture', in *ATAL*, volume 1757 of *LNCS*, pp. 364–378, (1999).
[3] Frank Dignum, David N. Morley, Liz Sonenberg, and Lawrence Cavedon, 'Towards socially sophisticated BDI agents', in *ICMAS*, pp. 111–118. IEEE Computer Society, (2000).
[4] Davide Grossi, Huib Aldewereld, and Frank Dignum, 'Ubi Lex, Ibi Poena: Designing Norm Enforcement in E-Institutions', in *COIN II*, volume 4386 of *LNCS*, pp. 101–114, Berlin, Heidelberg, (2006).