Norm Identification in Jason using a Bayesian Approach

Guilherme Krzisch** and Felipe Meneguzzi

School of Computer Science
Pontifical Catholic University of Rio Grande do Sul
Porto Alegre, Brazil

guilherme.krzisch@acad.pucrs.br,felipe.meneguzzi@pucrs.br

Abstract. Open multi-agent systems consist of a set of heterogeneous autonomous agents that can enter or leave the system at any time. As they are not necessarily from the same organization, they can have conflicting goals, which can lead them to execute conflicting actions. To prevent these conflicts from negatively impacting the system, a set of expected behaviors – which we refer to as *norms* – can desirable; to enforce compliance to such norms, sanctioning of violating agents can be used to deter further violations. As new agents enter the system, they must be able to identify existing norms in order to avoid sanctions. In this context, this paper provides two contributions. First, we propose a normative multi-agent system that can be used to evaluate norm-identification algorithms. Second, we validate an existing bayesian norm-identification approach in this system, confirming its positive result in a set of experiments.

Keywords: norm identification, normative system, multi-agent system

1 Introduction

Multi-agent systems allow the specification, modeling and implementation of complex behaviors generated by multiple autonomous agents interacting in a common environment. If these agents can perform actions that interfere with each other and jeopardize the overall functioning of the system, some kind of coordination mechanism can be employed to prevent this negative impact [5]; this can be achieved using regimentation or enforcement approaches. This first approach restricts the possible actions of the agents by design, completely preventing forbidden actions. While regimentation precludes violations, it also decreases the agent autonomy (e.g. in [6]). The latter, in turn, enforces a set of desirable behaviors (norms) by sanctioning violating agents (e.g. in [3, 7, 9, 20]). This has two main advantages: it allows agents to reason whether to follow a norm-compliant or a norm-violation behavior based on, for example, its resulting

** This work is partially supported by grant from CNPq/Brazil (132339/2016-1).

expected utility, and it enables an open multi-agent system where agents are not necessarily designed by the same organization [5].

As the expected behavior is not known at design time in enforcement approaches, the participating agents must be able to identify norms currently being enforced in a given system. This can be necessary, for example, in systems in which norms are not explicitly available or if there is no trust between agents. There are many different approaches to norm identification in the literature [17, 18, 12, 2, 1, 11]. In this paper we leverage an existing Bayesian approach [4] to develop a norm identification procedure within an agent simulation [8]. In order to validate the resulting approach we propose a normative multi-agent system testbed. We perform a set of experiments using this testbed; the results show that the employed approach is able to correctly identify the existing norms in the system, enabling agents to start taking into account these norms in its reasoning process, and thus allowing them to avoid sanctions.

2 Background

In this section we describe the Jason platform which is used to develop the multi-agent system, and its companion CArtAgO to implement artifacts which can be manipulated by agents. Then we describe how we formalize norms and how it relates to the Bayesian norm identification approach.

2.1 Jason with CArtAgO

Jason is based on the AgentSpeak language [13], which in turn implements the BDI architecture (belief, desires and intentions) [14] to simulate agent reasoning. An agent designer provides a set of plan-rules to achieve an implicit goal; these plans are chosen based on the current context of the agent beliefs and the set of available plans for an agent is called the *plan library*.

While Jason provides a framework for the internal reasoning of the agents, CArtAgO (Common ARTifact infrastructure for AGents Open environments) provides the abstraction of a virtual environment [15] in terms of artifacts. Artifacts contain a set of operations available to agents and are a useful abstraction of components used to perform a certain coordinated behavior among agents.

2.2 Norms

Norms exist in a society and are used to define the expected behavior of agents when performing actions in this environment [10]. Their function is to avoid potential harmful behavior that negatively impacts society, e.g. agents driving on the left and on the right side of the road, as this would lead to a high number of car accidents. Norms can be violated by individual agents if they reason that this is the best course of action, i.e. if an agent reasons that the outcome of a norm violating behavior is more desirable than compliance. This makes norms more flexible than hard-constraint rules specified at design time,

and over which agents have no choice, limiting their autonomous behavior. As a norm can be violated, it must be enforced in order to remain active, i.e. agents not following established norms must be sanctioned to deter further violations; this enforcement can be carried out by an authoritative organization or by other agents in the society [16].

According to [16], there are five phases of norm development: creation, identification, spreading, enforcement and emergence. In the current work we focus on the norm identification phase, which refers to the problem of how new agents entering the society can infer the norms created and currently being enforced in the system. We implement and validate a recent approach proposed in the literature, which uses the Bayes Theorem to make this inference, described in the next section.

2.3 Norm identification using a Bayesian approach

In this section we describe a norm identification approach which uses the Bayes Theorem in order to infer a set of norms in a given society [4]; we refer to the original paper to more detailed information. Norm identification approaches usually infer whether a norm exists in the society by looking at the actions performed by existing agents in the system. For this, such approaches assume that they have a model of how the system works and that they can collect a set of observations; the first can be encoded as a state-space graph of the possible transitions in the system, where nodes are states and edges are agent actions, while the second is a list of observations, where each one is a sequence of nodes visited by an existing agent.

In this approach, norms are defined in a subset of linear temporal logic (LTL), which specifies constraints on sequences of states. They can be either obligations (*eventually* or *next*) or prohibitions (*never* or *not next*); having the following six norm interpretations:

1. $eventually(\delta)$: Constrain a plan execution to include node δ.
2. $never(\delta)$: Constrain a plan execution to exclude node δ.
3. $next(\gamma, \delta)$: Constrain a plan execution to, when agent reaches context node γ, include node δ, where exists an edge from γ to δ in the graph.
4. $not\_next(\gamma, \delta)$: Constrain a plan execution to, when agent reaches context node γ, exclude node δ, where exists an edge from γ to δ in the graph.
5. $eventually(\gamma, \delta)$: Similar to item 3, but it is not necessary to exist an edge from γ to δ. This indicates that node δ will eventually be reached from node γ.
6. $never(\gamma, \delta)$: Similar to item 4, but it is not necessary to exist an edge from γ to δ. This indicates that node δ will never be reached from node γ.

Given the above six norm interpretations, there are a number of possible norm hypotheses with respect to a state-space graph; all these possible norm hypotheses are candidates for actual norms in the system. The norm hypotheses are weighted according to a number of observations given by some new agent

in the system; each observation contains a sequence of states in the state-space graph, executed by existing agents.

The approach we employ [4] uses an alternative interpretation of the Bayes Theorem that computes the odds of each possible norm hypotheses against a null hypothesis (i.e. the hypothesis that there are no norms), given some observed data D:

$$O(H_1 : H_2|D) = \frac{p(H_1|D)}{p(H_2|D)} = \frac{p(H_1)p(D|H_1)/p(D)}{p(H_2)p(D|H_2)/p(D)} = O(H_1 : H_2)\frac{p(D|H_1)}{p(D|H_2)}$$

, where H are the set of hypotheses and $O(H_1 : H_2)$ is the prior odd of H_1 over H_2. The prior odds of the null hypothesis is defined as one, while for the other norm hypotheses is set to an arbitrary value less than one. Note that here, each norm is considered in isolation against a null hypothesis of there being no norm. The candidate norms became actual norms when their relative odds is greater than the odds of other norm hypotheses. We refer to the original paper for further details, and in the following sections we describe the scenario and experiments performed.

3 Norm-detecting system

We developed a multi-agent system testbed in Jason with CArtAgO. The environment is a park (based on [19]), where agents can move in a grid simulating a park environment. There are bars where agents can buy food or beverages; after that, they can act in two ways: they can go to a trash can to recycle the waste or they can discard it somewhere in the park. In the first case they are *non littering agents* and in the second they are *littering agents*. Agents perform these actions and walk randomly in the park until the simulation ends. In this system, a norm is established when almost every single agent is from the same type, i.e. *littering* or *non littering*.

Figure 1 shows a park environment example. The trash can is located at the top left, in gray, and the bar is at the center, in green; yellow diamonds represent garbage in the environment, and agents are represented by circles (dark and light blue circles represent *non littering agents* carrying or not litter; gray and black circles represent *littering* agents carrying or not litter).

All agents start with a score of 100 utility points, being either *littering* or *non littering* agents, which we refer to as their *strategy*. The agents change their strategy once its score reaches a certain threshold; in the current work we set this threshold to 50 utility points. There are two sources of change in this score: the first is when they litter or when they recycle; in the first case they have a gain of utility of 0.5 points, while in the latter case they loss 0.5 points of utility. These values represent the fact that is easier to litter than it is to find a trash can and recycle.

The second source of change in the agent scores is when a *non littering agent* observes another agent littering. This can occur when both agents are within an observing distance of one another, i.e. agents cannot observe all other

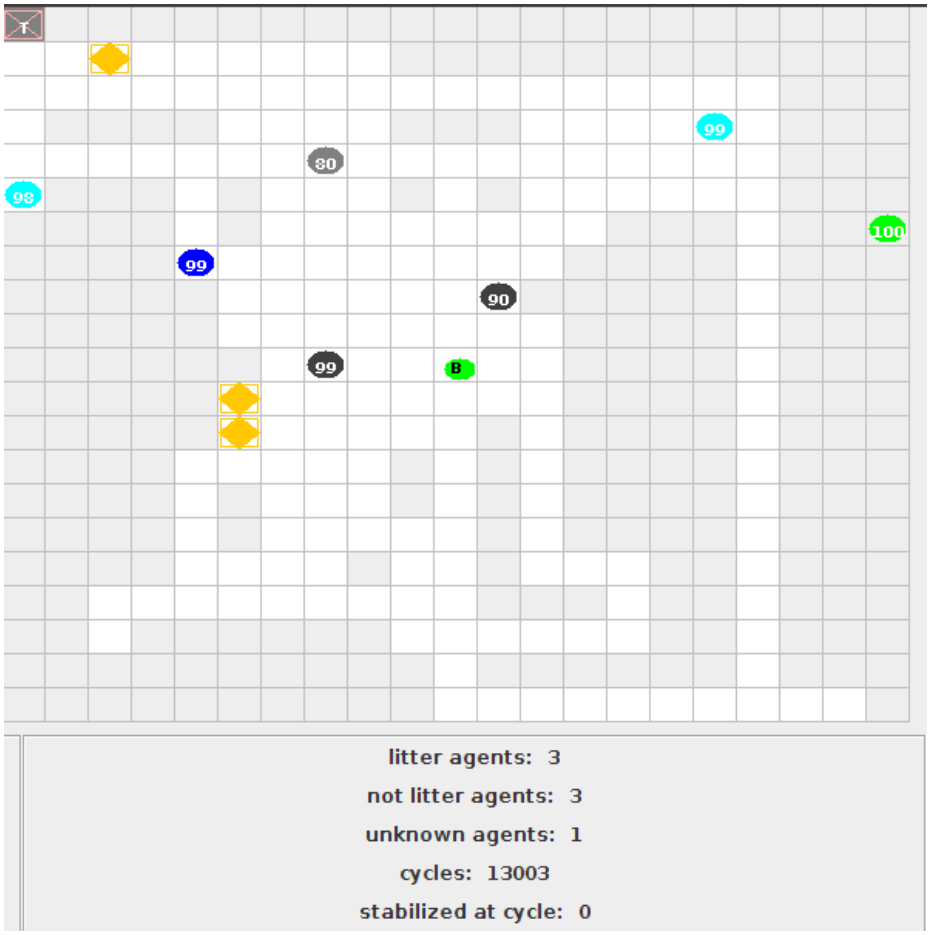

Fig. 1: Example of a park environment with seven agents

agents and their performed actions in the environment. In this situation, the observer agent *yells* at the other agent, losing a very small enforcement cost (0.01 of utility); consequently, the agent that littered loses 10 utility points from its score, representing a reputation loss or some loss derived from a negative emotion (e.g. guilty).

When a new agent enters in the environment, it collects observations to infer the current norms. In order to do this, it first needs a representation of the state-space of the possible states and actions in this system. Figure 2 shows a possible representation, where nodes are states and edges are actions available to the agents. Note that not all actions present in the plan library appear in this graph for readability purposes; we omit irrelevant actions (which will not give us any useful information of the existing norms) in the figure only (but they are represented internally in the agents), like recursive actions that try to move from

one location to another. In the figure, states are labeled as a single character with their corresponding description inside parentheses; as input to the norm identification algorithm we will provide just the single characters.

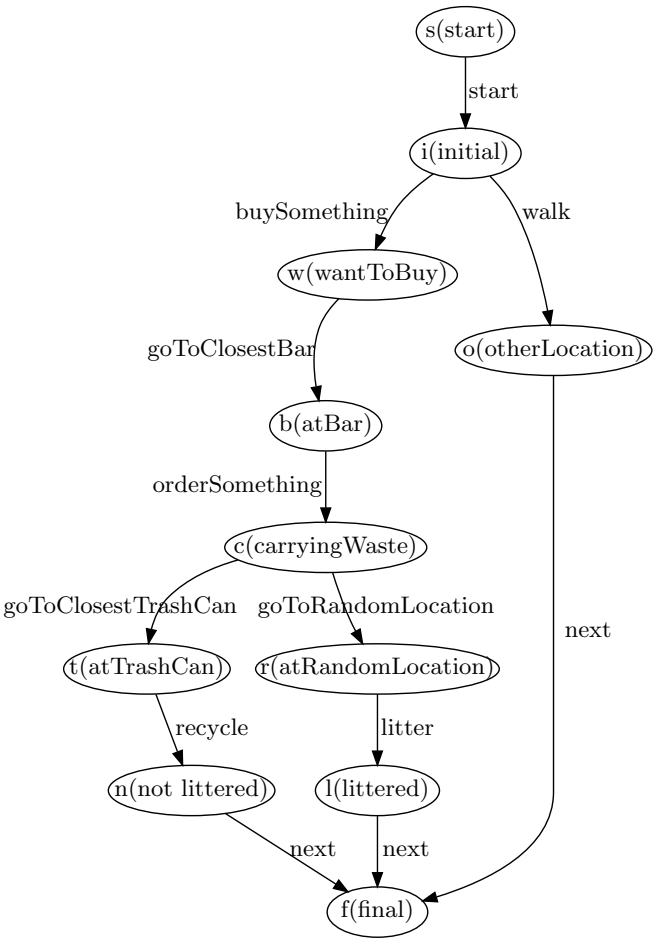

Fig. 2: State-space graph of the plan library for an agent in the park environment

Having the representation of the state-space graph, we now describe the procedure to infer the established norms, shown in Algorithm 1. It starts with the agent collecting a set of observations, where each observation is a sequence of characters in the graph (Line 2). The algorithm then provides this set of observations as input to the Bayesian norm identification algorithm (Line 3), which in turn calculates the odds of all norm hypotheses; as these odds are not absolute and must be considered as relative to other norm hypotheses, we only retrieve the ten most probable norm hypotheses to infer the current norm

(Line 4). We filter these ten norm hypotheses to detect the relevant norm to our problem (Line 5), i.e. if there is a norm to *litter* or to *not litter*. To perform this filter, we are interested in norm hypotheses where δ is t, n, r or l – i.e. the main nodes in the graph that discriminates between the two behavior we are interested in. For the norm interpretations where there is a node γ, we filter those that are i, w, b or c – i.e. the nodes in the graph which contains a path to nodes in δ. We then perform further processing to check which is the most probable norm based on the corresponding norm hypotheses relation (Line 7):

1. for *next* or *eventually*: if $\delta = (t$ or $n)$, then this is an indication that a *not litter* norm is present in the system; if $\delta = (r$ or $l)$ it is an indication of a *litter* norm.
2. for *not next* or *never*: this is the opposite of the above rule, e.g. if $\delta = (r$ or $l)$, then this is an indication that a *not litter* norm exists in the system.

The new agent in the society adopts the behavior of the most probable norm, based on the number of indications of the *litter* and *not litter* norm; for this, it chooses the norm with the highest number of indications (Line 9). In case of a draw, the agent can either keep collecting observations until it infers a norm or it can arbitrarily adopts a norm (e.g. a *not litter* norm).

Algorithm 1 Norm Inference Procedure

```
1: procedure NormInferenceProcedure(stateSpaceGraph)
2:     observations ← collect a set of observations
3:     normHypotheses ← normIdentificationAlgorithm(stateSpaceGraph, observations)
4:     topTenNormHypotheses ← retrieve top ten hypotheses from normHypotheses
5:     filtered ← filter relevant topTenNormHypotheses
6:     for normHypothesis in filtered do
7:         check if normHypothesis indicates a litter or not litter norm
8:     end for
9:     return most probable norm based on the number of each norm indications
10: end procedure
```

4 Experiments and Results

In order to evaluate the accuracy of correct identification of existing norms, we ran a set of simulations on the environment described in the previous section. More specifically, we added a new agent in the system that collects a set of observations over 10000 execution cycles; this results in an average of two observations for each observable agent in the system.

We designed four different types of experiments: the first one is designed to test the accuracy of new agents detecting a *not litter* norm, while in the second

experiment there is a *litter* norm. In the third experiment there is no established norm in the society; finally, in the last experiment we test the accuracy in relation to the number of existing agents in the system.

4.1 Not Litter Norm

For the first experiment we simulate the environment with six existing agents, where all agents are of the *non littering* type, thus this society has an established *not litter* norm. We add a new agent in the system, which collects a set of observations; a sample of a set of observations follows:

1. i, o, f
2. i, w, b, c
3. i, w, b, c, t, n, f

These sequences can be partial in the state-space graph of the scenario, i.e. they do not need to begin in the initial node state and finish in the end node state, because agents have a limited observing time and can only observe a limited set of agents which are at a close distance. From the first observation we cannot infer any norm, because this is a sequence of states of an agent that decided to randomly walk in the park. The second observation represents a partial sequence of states which ends with the state where the agent is "carrying waste"; again, this does not indicate any norm. Finally, the third observation indicates that a *not litter* norm exists, because it is a sequence of states of an agent that has recycled its waste.

An example of the output of the Bayesian norm identification algorithm given this setup is the following top ten norm hypotheses:

1. ('c', 'next', 't')
2. ('c', 'not next', 'r')
3. ('c', 'eventually', 'n')
4. ('l', 'not next', 'f')
5. ('c', 'never', 'r')
6. ('r', 'never', 'l')
7. ('r', 'eventually', 'i')
8. ('l', 'eventually', 's')
9. ('l', 'eventually', 'n')
10. ('r', 'eventually', 'c')

From these hypotheses, we can infer that a *not litter* norm exists. This is supported by: the first norm hypothesis, indicating that after an agent is in node c, it will go to node t (it will recycle); the second norm hypothesis, indicating that agent will not go to a random location to litter; the third norm hypothesis, indicating that agent will eventually recycle; and the fifth norm hypothesis, indicating that agent will never go to a random location to litter. All other hypotheses are irrelevant for the detection of the existing norm. For this experiment, all simulations correctly inferred the *not litter* norm; this enables the new agent to adopt the established norm.

4.2 Litter

This experiment is similar to the previous one, but instead of an existing *not litter* norm, there is a *litter* norm established in the society. An example of the top ten norm hypotheses follows:

1. ('w', 'eventually', 'l')
2. ('n', 'eventually', 'c')
3. ('t', 'not next', 'n')
4. ('n', 'eventually', 'r')
5. ('t', 'eventually', 'o')
6. ('b', 'never', 'n')
7. ('n', 'eventually', 'n')
8. ('i', 'never', 'n')
9. ('never', 'n')
10. ('w', 'never', 't')

Norm hypotheses one, six, eight, nine and ten indicate that there is a *litter* norm, because they lead us to nodes r and l and away from nodes t and n. Running this experiment in a set of simulations resulted in all new agents being able to correctly infer the existing *litter* norm.

4.3 Undefined

While the previous experiments have an established norm, in this experiment we have half *littering* agents and half *non littering* agents; the expected result is that the new agent will not be able to infer any norm. An example of the top norm hypotheses follows:

1. None
2. ('w', 'never', 'w')
3. ('s', 'never', 'n')
4. ('s', 'eventually', 'n')
5. ('l', 'never', 's')
6. ('s', 'never', 'l')
7. ('b', 'never', 's')
8. ('t', 'never', 'l')
9. ('n', 'next', 'f')
10. ('w', 'never', 's')
11. ('f', 'never', 'o')

None of these norm hypotheses indicate that there is an established norm. Accordingly, in the set of simulations the new agents were (correctly) not able to infer any norm.

4.4 Increasing the number of agents

For the last experiment we validate the norm identification approach on a society with an increasingly large number of agents. We perform several simulations, and in all cases where all existing agents in the society have a *not litter* norm or where all have a *litter* norm, the new agent was able to correctly infer the established norm.

When the relation between the number of *non littering* and *littering* agents is close to one, and therefore there is no norm currently established, the approach correctly infers so. When this relation is disproportional, i.e. there are many more agents of one type than of the other, the approach is also capable of inferring the norm of the predominant type. For example, with 50 agents, 95% *non littering* agents and the remaining 5% *littering* agents, the approach inferred a *not litter* norm. With 100 agents, 90% *littering* agents, the *litter* norm was inferred.

Table 1 shows results from experiments with an increasingly number of agents, changing the relation between *non littering* and *littering* agents, along with its corresponding inferred norm. When the approach is not able to infer any norm, the new agent being added to the society can either assume an arbitrary norm or can keep collecting observations.

Percentage of *littering agents*	# of agents	Inferred norm
100% to 90%	6	litter
	50	litter
	100	litter
85% to 10%	6	none
	50	none
	100	none
5% to 0%	6	not litter
	50	not litter
	100	not litter

Table 1: Inferred norms for an increasingly number of agents, and the percentage of *littering* and *non littering* agents

5 Conclusion and Future Work

In this paper we described an experiment to validate a norm identification approach in a multi-agent system implemented in Jason with CArtAgO. More specifically, we used a Bayesian norm identification from [4] as a base to get the most probable norm hypotheses, and then process these results to infer if there is a norm established in the society. This paper provides two main contributions. First, we developed a norm inference testbed in a popular agent programming language that can be used for experiments of norm-identification algorithms.

Second, we have conducted further experiments to validate the bayesian norm-identification approach by Cranefield *et al.* [4], confirming their positive result in a multi-agent setting.

In order for the Bayesian norm identification approach classify the norm hypotheses, it needs both the state-space graph of the problem and a set of observations. We manually built the state-space graph of the problem, identifying its key states and actions. For future work we intend to try to automatically generate the state-space graph of the plan library built in Jason; the main challenges would be to identify the key components of the problem and to remove loops which exists inside the plan library. This would allow the Bayesian norm identification approach to be applied to any system built in Jason.

We also intend to run more experiments in different and more complex scenarios, with norms with increasing complexity, to further evaluate the employed approach. We would also like to investigate different ways of combining the top norm hypotheses, maybe introducing weights accordingly to their relative odds.

References

1. Alrawagfeh, W., Brown, E., Mata-Montero, M.: Norms of behaviour and their identification and verification in open multi-agent societies. In: Theoretical and Practical Frameworks for Agent-Based Systems. IGI Global (2012) 129–145
2. Andrighetto, G., Conte, R., Turrini, P., Paolucci, M.: Emergence in the loop: Simulating the two way dynamics of norm innovation. In: Dagstuhl Seminar Proceedings, Schloss Dagstuhl-Leibniz-Zentrum für Informatik (2007)
3. Boella, G., Van Der Torre, L., Verhagen, H.: Introduction to normative multiagent systems. Computational & Mathematical Organization Theory **12**(2-3) (2006) 71–79
4. Cranefield, S., Savarimuthu, T., Meneguzzi, F., Oren, N.: A bayesian approach to norm identification. In: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, International Foundation for Autonomous Agents and Multiagent Systems (2015) 1743–1744
5. Dignum, F.: Autonomous agents with norms. Artificial Intelligence and Law **7**(1) (1999) 69–79
6. Esteva, M., Rosell, B., Rodriguez-Aguilar, J.A., Arcos, J.L.: Ameli: An agent-based middleware for electronic institutions. In: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 1, IEEE Computer Society (2004) 236–243
7. García-Camino, A., Rodríguez-Aguilar, J.A., Sierra, C., Vasconcelos, W.: Constraint rule-based programming of norms for electronic institutions. Autonomous Agents and Multiagent Systems **18**(1) (2009) 186–217
8. Krzisch, G., Meneguzzi, F.: Norm Identification in Jason using a Bayesian Approach. https://doi.org/10.5281/zenodo.438046 (March 2017) [Online; accessed 24-March-2017].
9. Luck, M., d'Inverno, M., et al.: Constraining autonomy through norms. In: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems. (2002) 674–681
10. Luck, M., Mahmoud, S., Meneguzzi, F., Kollingbaum, M., Norman, T.J., Criado, N., Fagundes, M.S.: Normative agents. In: Agreement technologies. Springer (2013) 209–220

11. Mahmoud, M.A., Ahmad, M.S., Ahmad, A., Yusoff, M.Z.M., Mustapha, A.: The semantics of norms mining in multi-agent systems. In: International Conference on Computational Collective Intelligence, Springer (2012) 425–435
12. Oren, N., Meneguzzi, F.: Norm identification through plan recognition. In: Proceedings of the workshop on Coordination, Organization, Institutions and Norms in Agent Systems (COIN 2013@ AAMAS). (2013)
13. Rao, A.S.: Agentspeak (l): Bdi agents speak out in a logical computable language. In: European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Springer (1996) 42–55
14. Rao, A.S., Georgeff, M.P., et al.: Bdi agents: From theory to practice. In: ICMAS. Volume 95. (1995) 312–319
15. Ricci, A., Viroli, M., Omicini, A.: Cartago: A framework for prototyping artifact-based environments in mas. In: Environments for Multi-Agent Systems III. Springer (2006) 67–86
16. Savarimuthu, B.T.R., Cranefield, S.: Norm creation, spreading and emergence: A survey of simulation models of norms in multi-agent systems. Multiagent and Grid Systems **7**(1) (2011) 21–54
17. Savarimuthu, B.T.R., Cranefield, S., Purvis, M.A., Purvis, M.K.: Obligation norm identification in agent societies. Journal of Artificial Societies and Social Simulation **13**(4) (2010) 3
18. Savarimuthu, B.T.R., Cranefield, S., Purvis, M.A., Purvis, M.K.: Identifying prohibition norms in agent societies. Artificial intelligence and law **21**(1) (2013) 1–46
19. Savarimuthu, B.T.R., Purvis, M., Purvis, M., Cranefield, S.: Social norm emergence in virtual agent societies. In: Declarative Agent Languages and Technologies VI. Springer (2008) 18–28
20. Sierra, A.P.d.P.C., Schorlemmer, M.: Friends no more: Norm enforcement in multi-agent systems. (2007)