AVOIDANCE OF NORM VIOLATION IN MULTI-AGENT ORGANIZATIONS

Charlotte Gerritsen and Mark Hoogendoorn
Vrije Universiteit Amsterdam
Department of Artificial Intelligence
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
{cg, mhoogen}@few.vu.nl
http://www.few.vu.nl/~{cg, mhoogen}

KEYWORDS
Multi-agent organization modeling, norm violation, avoidance

ABSTRACT

In contemporary society not adhering to norms is something which is unwanted. Currently approaches to prevent this from happening are often taken whereby the deviation of a norm is punished and possible repair actions are performed. However, this is a reactive approach and can only happen once the norm has already been violated. This paper presents a proactive approach which allows intervention before the deviation actually occurs. In order to do this, an approach is specified for agents that enforce norms and can influence the input states of agents. This approach includes learning of input/output correlations of these agents by constructing decision trees, and utilizing this decision tree to intervene such that norm violation can be avoided. The approach is evaluated in the domain of Criminology.

INTRODUCTION

In order for agent societies to function in an effective and structured way, the paradigm of multi-agent organizations and institutions is a perspective which has been used by a variety of authors (see e.g. Ferber and Gutknecht 1998; Hannoun et al 2000). In such approaches, societies are specified by means of a structure (for instance by means of roles), as well as the expected behavior of such an organization. Norms (see e.g. Castelfranchi 2000) can be used to specify this expected behavior. A crucial point of specifying this behavior is that the agents participating in the society comply with these norms, as violation of these norms can potentially be harmful for the society as a whole. Hereby, several approaches have been developed to detect and punish these violations by using enforcer agents, possibly repairing the consequences (see e.g. Vazquez-Salceda et al 2004; Boella and Van der Torre 2008).

When looking at human societies a variety of organizations is present that enforce norms and punish those who violate the norms e.g. the judicature and the police. These efforts are however not limited to punishing the society members that have already violated the norms, but also efforts are performed to avoid these violations.

This paper uses the idea to prevent norm violations from human society, and translates it to the domain of normative organizations. The approach is intended to support humans in preventing norm violation by being advised by an agent, or by letting the agent itself proactively intervene. The approach itself is presented in a highly generic format in order to make it reusable. Such a generic approach is however difficult to evaluate. In this paper an evaluation has therefore been conducted within a specific domain, namely Criminology, because this is a typical area in which norms are frequently violated. Criminology is a social scientific area which focuses on all aspects of criminal behavior. So on prevention as well as on punishment, making it very suitable as an application domain.

The approach is developed under the assumption that these advisory or norm avoidance agents can observe the input and output states of other agents participating in society and influence particular input states of agents. Ideally, these agents would simply avoid unwanted actions from occurring, but these might not all be controllable. Therefore, in the approach first of all algorithms are present for learning the input/output correlations of the agents in the society. This can be learned by either taking an external perspective, not knowing the internal functioning of an agent, or by having full knowledge of the internal functioning of this agent. This correlation can be used to predict whether the current input states of the agent lead to an unwanted output state. In case norm violation is predicted using these correlations, methods are introduced to avoid this violation from occurring by influencing the input states of the agent.

The paper is organized as follows. First, generic formalisms to avoid norm violation are presented in three parts: (1) a part that enables derivation of norm violation and representation of input and output states of an agent; (2) the learning of input/output correlations of agents, and (3) methods to avoid norm violation. Thereafter, the approach is thoroughly evaluated for a case within the domain of Criminology to see how well the system would function in such a domain. Finally, a discussion is presented.

Proceedings 23rd European Conference on Modelling and Simulation ©ECMS Javier Otamendi, Andrzej Bargiela, José Luis Montes, Luis Miguel Doncel Pedrera (Editors)
ISBN: 978-0-9553018-8-9 / ISBN: 978-0-9553018-9-6 (CD)

AGENTS AND NORM VIOLATION

This section discusses the perspective taken towards agents within a particular organization, and the norms that are specified to constrain the behavior of such agents. Note that the formalisms introduced here for agents and norm violation are meant as a way to specify the approach to prevent violation of norms, the idea is therefore to introduce a very simple notation to allow a clear explanation of the approach. Of course, much more advanced formalisms can be used as well, but that is not the aim of this paper.

Viewpoint Towards Agents

The *external viewpoint* towards agents is represented by means of input and output states of the agent. An input of an agent a can be built up using elements from the set of possible *observations* the agent can receive, expressed by the ontology $O_{in(a)}$. In addition, an input state of an agent can consist of *communication*. Hereby the ontology of communications an agent can receive is represented in the set $C_{in(a)}$. Output states of an agent can include *communication* ($C_{out(a)}$) or *actions* ($A_{out(a)}$). Together these sets form the ontology for external states:

$Ont(a) = O_{in(a)} \cup C_{in(a)} \cup C_{out(a)} \cup A_{out(a)}$

In order to derive what is true in the world, truth values are assigned to each of the atoms that can be created using the set Ont(a). This combination is referred to as the set Ext(a), with which the actual occurrence of external states of the agent can be described. The occurrence of the truth values of these elements varies over time. The dependencies between elements are represented in temporal logic. Furthermore, it is assumed that a *history* of the agent can be maintained by storing previous states of the agent. Hereby a previous state is made explicit by means of the prev predicate (which can be nested up to s states back in time), which can be built as follows:

$PrevState(a) = prev^n(Ext(a))$ where $1 \leq n \leq s$ and

$prev^1(Ext(a)) = prev(Ext(a))$

Besides an external viewpoint towards an agent, more might be known about the *internal processes* of the agent as well. Hereby, internal states can become necessary, for example beliefs, desires and intentions (e.g. Rao and Georgeff, 1991). These states can be expressed in a similar fashion as the input and output. Dependencies between the different states can be represented by using a temporal logic. For example, an observation o_1 can lead to an action a_1 being performed in the next state, in case this is true, this can be specified in temporal logic as follows: $o_1, \rightarrow \circ\, a_1$

Norm Violation

As already argued in the introduction, violation of norms by agents is unwanted. Several mechanisms for detection of norm violation have been proposed (see e.g. Boella and Van der Torre 2008). The assumption in this paper is that given a set of external agent states it can be determined whether a norm is violated. The focus is on the single agent case, therefore, the norms addressed do not concern multiple agents. Assume a certain norm n_1 from the set of norms for an agent a called N_a (i.e. $n_1 \in N_a$). A subset of the current inputs and outputs of the agent and the input and outputs that occurred in the past can determine whether such a norm will be violated at some point in time in the future (i.e. the norm does not hold at that time point):

$NormViolationSt(a, n_a) \rightarrow \Diamond \neg n_a$

where $NormViolationSt(a, n_a) \subseteq (Ext\ (a) \cup PrevState(a))$

Hereby, the condition is that this set is minimal:

$\neg\exists NormViolationSt2(a, n_a)$:

$NormViolationSt2(a, n_a) \rightarrow \Diamond \neg n_a \wedge$

$NormViolationSt2(a, n_a) \subset NormViolationSt(a, n_a)$

For instance, if the norm n_1 states that action a_1 should never be performed, the fact that the agent performs the action makes that this norm is violated: $a_1 \rightarrow \Diamond \neg n_1$

IDENTIFYING POTENTIAL NORM VIOLATION

This section addresses how potential violation of norms can be detected, for instance by an advisory agent for a human. Hereby, a distinction is made between the case whereby a complete internal model of the agent is present, and the case whereby only an externally observable history of the agent is present (i.e. the input and output states).

Externally Observable History

In order to identify the states of the agent that lead to the occurrence of a norm violation state, it is assumed that the advisor agent can observe the external states of an agent at all times. See the last section for a brief discussion on this assumption. In order to learn the correlation that exists between the occurrence of these external states and the set which implies violation of a norm, a data mining algorithm is used. More in specific, the ID3 algorithm as proposed by Quinlan (1979). This algorithm can be used to construct a decision tree based upon which it can be determined whether the recent history and present states of the agent imply the future occurrence of the norm violating set.

The algorithm works as follows: an attribute (in this case the attributes are states of the agent) is chosen which minimizes the function that calculates the information entropy:

$$E(A) = -\sum_{i=1}^{V} \frac{S_i}{S} \sum_{j=1}^{N} \frac{k_{ji}}{S_i} \log_2 \frac{k_{ji}}{S_i}$$

Whereby V is the number of values for attribute A, (in this case 2, true and false), S is the total number of examples, and S_i is the total number of examples where A has the i^{th} value. In this specific case, the examples concern the input and output states of the agent that are not present within the $NormViolationSt(a, n_a)$ set, and the occurrence of the specific set $NormViolationSt(a, n_a)$ which is the classification of the examples (i.e. whether the norm will be violated or not). Furthermore, the history which is taken into consideration is limited (it is not the aim to take the complete history into account). Therefore, the number of previous states is limited to δ:

IndicationViolation(a, n_a) $\subseteq$

(Ext (a) $\cup$ prev(Ext(a) $\cup$ .... $\cup$ $prev^{\delta}$(Ext(a))) where

IndicationViolation(a, n_a) $\cap$ NormViolationSt(a, n_a) = $\varnothing$

N in the equation to calculate the entropy is the number of categories for the classification. In this case, it is always equal to 2 since a norm violation state can occur or not. Finally, k_{ji} is the number of examples in the j^{th} category of the classification that have the i^{th} attribute value. Given that such an attribute is chosen based upon this function, the examples are partitioned according to the attribute values of the chosen attribute. This process is continued iteratively, unless the set of examples consists of merely one category for classification. Then the leaf is simply labeled with that category. In addition, several mechanisms can be used to avoid overfitting the data, such as *chi-squared* pruning (Quinlan 1979).

Imagine the very simple scenario whereby the set NormViolationSt(a, n_1) consists of one element, namely a_1, and a_1 is observed to be the case in the current state. Furthermore, the only relevant history is the previous state, in which o_1 was true. Then, the following input will be presented to the ID3 algorithm (the states are in the top row of the table and the accompanying truth values in the second row):

prev(o_1)	NormViolationSt(a, n_1)
true	true

The resulting decision tree will be trivial in this case, namely just 'true'.

Complete Internal Model

In case a complete internal model is present of the agent (this is for instance common in Criminology in case a detailed psychiatric report is present), a decision tree can be constructed as well based upon the external states. The underlying assumption is that the values of the internal states of the agent that do not depend upon external states are known, and do not change over time. The external states considered in the decision tree are the current states, and the history up till δ steps back. This set is defined as follows:

RelevantStates(a) = Ext (a) $\cup$ prev(Ext(a) $\cup$ .... $\cup$ $prev^{\delta}$(Ext(a))

Given this set, a decision tree can now be constructed to indicate a certain norm violation set NormViolationSt(a, n_a) occurring as specified in algorithm 1 below. Note that the algorithms are not presented in a full formal format for the sake of clarity.

Algorithm 1. Constructing Decision Tree using Internal Model

Start by constructing a root node of the tree which represents the state without the truth value in the first element in the set RelevantStates(a). Construct two branches labeled true and false. Set i=2

1. if i $\leq$ |RelevantStates(a)| For each branch without a node attached: construct a node labeled with the state without the truth value of the i^{th} element in RelevantStates(a).
   Construct two branches labeled true and false for each of these nodes, and set i = i + 1 and return to 1.
   else use the ontological elements and the truth values from the path from source to this leaf, and the internal model (including the values of states that are not dependent upon external states) to determine (by means of inference) whether these values result in at least one possible state that indicates the violation of a norm (i.e. NormViolationSt(a, n_a)) holds).
   if this is true, set this leaf to NormViolationSt(a, n_a) being true
   else set NormViolationSt(a, n_a) to false

As a result, a decision tree consisting of all (external) states, and a partial history, is constructed by means of which it can be determined whether a norm will be violated.

AVOIDANCE OF NORM VIOLATION

Given that it can now be predicted whether a norm will be violated, preventing this norm from being violated is the next step. In order to achieve this, the advisor agent needs advise the human how to change the world states to alter the input states of an agent (or proactively perform the changes itself). Since both the internal as well as the external norm violation model result in a decision tree, such a tree is taken as a basis. In this step, it is assumed that the violation of a norm is always more costly than the costs associated with avoiding norm violation.

In case the decision tree constructed expresses that a potential violation will occur (i.e. the tree predicts a violation) an intervention in the world can take place. The question of course is how to intervene. The basic idea is that only the input side of the agent can be influenced (for instance by sending a communication, or changing an observation). Hereby, an additional assumption is made, namely that the outcome of the reasoning process of the agent can still be influenced for a certain duration β by changing the input states. This allows the modification of the inputs of the agent during this interval β (given that transporting the new world information to the agent takes no time) and thus avoid the norm being violated. Deriving what input states to modify can be done by modifying the decision tree which has been constructed. This modification can be done by using the following algorithm:

Algorithm 2. Modifiable State Pruning

Start with X = source node

1. if X is not a leaf go to 2, otherwise if all nodes have been passed, go to 3.
2. if node X is an element which is not modifiable, and this element is either true or false, then you remove the entire branch representing the truth value which is not observed. Follow the non-removed branch to the next node, set X to that node, and return to 1.
   else if node X is modifiable, go to 1 for both the node when following the true and the node following the false branch.
3. from all leafs that are left and labeled true (i.e. the decision tree indicates this branch resulting in violation of the norm) start with Y = parent node of this leaf and do the following:

a. if Y only has one branch, remove the node and the branch that points towards this node, set Y to the parent node and go to a.
   else if Y has multiple branches, remove the branch that has just been taken.

If the result of this algorithm is an empty tree, it is unavoidable that the norm will be violated. Otherwise, the world can be modified such that the set of input states of the agent is identical to the set of states still present in the decision tree. The definition of whether an element is modifiable can be expressed as follows:

$$\text{modifiable}(E) \rightarrow$$
$$E \notin \text{PrevState}(a) \wedge E \in \text{Ext}(a) \wedge \text{modification\_time}(E) < \beta$$

Thus, an element is modifiable in case it is not part of a state in the past, and furthermore the modification time is shorter than the parameter β which has just been introduced. Figure 1 shows an example of a proof tree which has resulted from the ID3 algorithm. Hereby, the nodes represent the state name and the branches the truth value of this state.

Furthermore, it is given that both o_1 and o_3 in this tree are modifiable within duration β. In addition, it is given that prev(o_2) is false, o_1 is true, and o_3 is true. The resulting tree after application of algorithm 2 is shown in Figure 2.

The sets of states that can be reached and avoid violation according to this tree (i.e. all possible paths to the leafs of the tree) are referred to as AvoidViolation$_1$(a, n_a), ..., AvoidViolation$_n$(a, n_a). If n is greater than 1, one of the sets can be chosen based upon the costs of these modifications (whereby change_cost(s) indicates the cost to change input state s). Hereby, again an algorithm is specified for reducing the (already reduced) tree, in this case to one single option:

Algorithm 3. Costly State Pruning

For all nodes in the tree that still have multiple branches:
  Select the node X with the highest change cost and remove the entire branch of the truth value which is currently observed to be the case on the input state of the agent.

In the case of the tree shown in Figure 2, the following desired situation would result from application of algorithm 3: {o1, prev(o2), ¬o3} , in other words, only o3 needs to be modified from true to false. Hereby it does not matter what the cost are since only o_1 can be modified.

An assumption of algorithm 2 is that there is merely one violation, which might not always be the case. Therefore, the possibility to address multiple norms is addressed as well. Hereby, it is assumed that for each norm that plays a role in society, a decision tree is present. The idea is now to combine these trees into one tree, whereby the leafs represent which norms will be violated. In order to build such a tree, take the set of all nodes from the decision trees (i.e. IndicationViolation(a) = IndicationViolation(a, n_1) ∪ ... IndicationViolation(a, n_n)). Hereby the tree is constructed by iterating through all these elements as follows:

Algorithm 4. Creating Decision Tree for Multiple Norms

Start by constructing a source node of the tree which represents the ontological element in the first element of the set IndicationViolation(a). Construct two branches labeled true and false respectively. Set i to i=2

1. if i ≤ |IndicationViolation(a)| For each branch without a nodeattached: construct a node labeled with the state name without the truth value of the ith element in IndicationViolation(a). Construct two branches labeled true and false, and set i = i + 1 and return to 1.
   else construct leaf nodes in the tree that represent the entire set of norms {NormViolationState(a, n_1),..., NormViolationState(a, n_n)} under investigation. Label these by using the truth values of the different elements from the source of the tree to this leaf and the specific decision trees for the norms.

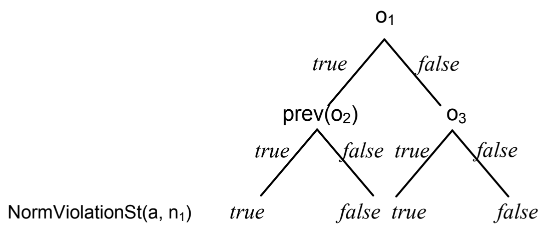

Figure 1. Example decision tree

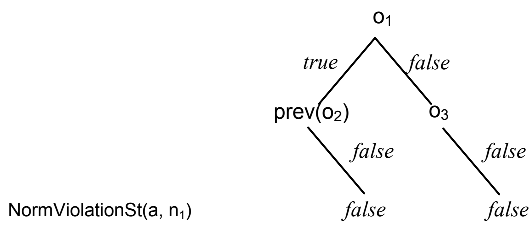

Figure 2. Pruned decision tree

In order to now prune the tree, the costs of norm violation are taken into account. Assumed that for each norm certain violation costs for society are specified (violation_cost(n_a)). The idea is to minimize the costs for society first, and in case multiple options are still feasible, select the option which costs least to manipulate. Given these costs the following pruning algorithm can be applied:

Algorithm 5. Pruning Decision Tree for Multiple Norms (extension of algorithm 2).

1. see algorithm 2.
2. see algorithm 2.
3. calculate for all leafs that are left in the tree, the costs of that leaf for the society (i.e. the sum of the costs of the norm violation sets that are labeled true in that leaf), and go to 4.
4. if there is more than one leaf left, select the leaf with the highest costs
   if the highest costs are greater than 0, start with Y = parent node of this leaf and do the following:
      a. if Y only has one branch, remove the node and the branch that points towards this node, set Y to the parent node and go to a
         else if Y has multiple branches, remove the branch that has just been taken, and go to 4.
   else go to 5.
   else go to 5.

5. if there is only one leaf left you are done
   else use algorithm 3 for pruning the remaining tree based upon the costs of changing an element.

CASE STUDY

In this Section, an evaluation is performed of the proposed approach within the domain of Criminology. Note that this evaluation has been conducted together with a criminologist. An important theory within Criminology, the routine activity theory (Cohen and Felson 1979), states that a crime will occur when a motivated offender meets a suitable target without a capable guardian being present. So to prevent crime from occurring one of the key aspects from the theory needs to be addressed. Punishing citizens that have violated norms can have a deterrent effect, and may alter the motivation of the offender. Another way to decrease the crime rates is to make sure that capable guardians are present. For example police surveillance, parking lot attendants or surveillance cameras and alarm systems. Further, one can try to make sure to not be a suitable target, so not to wear an obvious notebook case, or a wallet in your pocket. The problem is however that all offenders have different preferences. Some do not have high standards when it comes to their target and are not scared by the idea of being locked away in prison. What can be done about these hardcore offenders? The type of offender that is life persistent, and who seems to never be able to stay out of trouble? In this case the focus is on this type of offender.

The algorithms as introduced before have been implemented in Java. The outcomes of the algorithms for the specific scenario introduced below have been generated using the Java implementation. Note that this evaluation only addresses learning from history for the sake of brevity.

Table 1. Relevant input states of bank robber

Formal	Description	Modi-fiable
prev(presence_bank)	Bruce previously observed a bank being present	-
prev(quiet)	Bruce previously observed the bank located in a quiet area	-
prev(escape_route_outside)	Bruce previously observed a good escape route outside	-
less_than_6_people	Bruce observes less than 6 people inside the visible area of the bank	20
escape_route_inside	Bruce observes a good escape route inside the bank	5
surveillance_cameras	Bruce observes surveillance cameras in the bank	-
alarm_system	Bruce observes an activated alarm system (a red light blinking)	5
guardian	Bruce observes a person with a visible weapon in the visible area of the bank	15

Scenario Description

Bruce is a bank robber. The police have a large file on him, because he has been caught several times in the past. They have information on 20 attempts to rob a bank. However, not all of these banks actually got robbed. Bruce is known for always observing the bank for a while before he starts the robbery, usually at least 30 seconds. The police would like to know which aspects are relevant in his decision to rob a place. After going through his file they discover eight important input and previous input states (as shown in Table 1).

Table 1 describes the relevant input states, and the costs as calculated by the police of modifying the state. Hereby the previous states (observed outside the bank) are not modifiable. Note that these costs are not based on reality, but used as an example. Furthermore, some input states are not modifiable, e.g. the presence of surveillance_cameras is not modifiable since this can not be accomplished within 30 seconds. The less_than_6_people state can be modified by quickly moving people from/to the back office of the bank which is not visible for Bruce. Furthermore, the escape_route_inside is modifiable by visibly obstructing/opening the door. The alarm_system is modified by visibly switching on the blinking light of the alarm, or by switching it off. Finally, the guardian can be modified by the security guard or the personnel behind the desk hiding or showing a gun. Besides these input states there are two norms specified for this organization, namely robbery which specifies that the action of performing a bank robbery is not allowed, and shooting which specifies that shooting inside a bank is not allowed. In this case study, robbing a bank costs society 10,000, and shooting inside a bank (during a robbery) costs society 100,000.

Learning Based on History

After applying the ID3 algorithm for the robbery norm to the police history (using the full history as training set) the tree shown in Figure 3 results. Hereby, the branches represent the input states of Bruce whereas the leafs represent whether Bruce will rob (and thus violate the robbery norm) or not. The same process has been undertaken for the shooting norm, resulting in Figure 4.

Utilizing the Built Decision Tree

Due to his history Bruce got a parole officer assigned to him to track him during the day. One day the officer sees Bruce enter a bank. It is still early so there are not a lot of people on the streets yet. The bank lies near a highway entrance. So, once Bruce is outside the bank he can get away easily. The parole officer observes that there are 2 persons inside the bank and there is no guard present. The bank is equipped with surveillance cameras and an alarm system. Based on the trees built in the previous section, the advisor agent can derive that Bruce will rob the bank and use a gun to get away. After applying the algorithms introduced in the previous sections (algorithms 3-5) two input states are identified

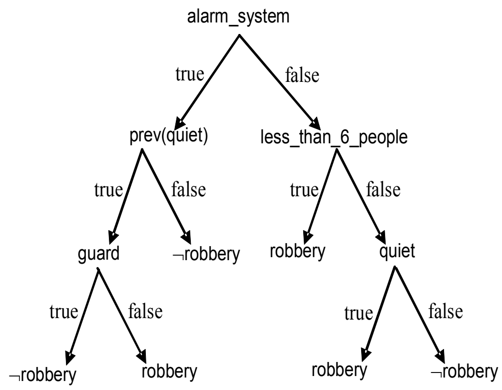

Figure 3. The tree for violation of the norm robbery

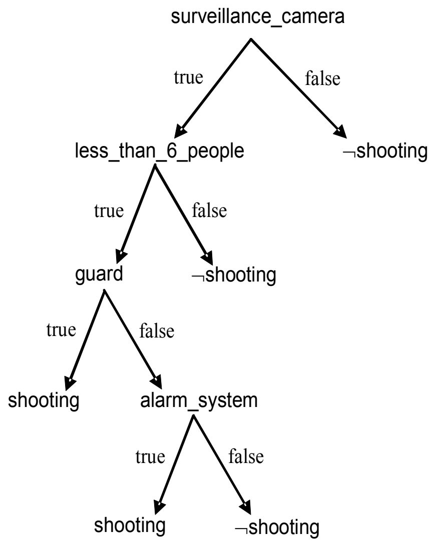

Figure 4. Decision tree for shooting norm

that need to be modified to avoid violation of the norms, namely making the state less_than_6_people false, and making the input state guard true. These changes are performed by the parole officer, and as a result, Bruce will not rob the bank, nor will he start shooting. According to the criminologist this result indeed complies with what was to be expected based upon the definition of the scenario.

DISCUSSION

In this paper it has been shown how potential norm violation can be detected. Norm violation can occur in a society in case the enforcers of these norms do not have full control over all actions of the agents in the society. Potential norm violation can be detected using a decision tree to forecast these violations. Two approaches to derive such a tree have been introduced: (1) learning from history using a data mining technique (in this case the ID3 algorithm), and (2) by building the decision tree based upon the internal model which is present of such an agent. The prediction information can be used in advisory agents, or even norm enforcing agents (e.g. police/parole officers). Under the assumption that these agents can receive the inputs of the other agents that will potentially violate a norm somewhat before they actually start processing, methods have been introduced that derive how this norm violation can be avoided. Such avoidance is accomplished by making adjustments in the world, and thus changing the input states of the agent. Several algorithms for deciding what states to change by looking at the costs of making this change and the costs to society have been specified as well.

The whole approach has been specified on a generic level, thus allowing reusability of the approach. The initial application of the approach presented in this paper is the domain of Criminology, in which it was shown to give accurate advice according to domain experts. An evaluation in other domains, for instance the domain of software agents, is future work.

One major issue within the approach presented is the scalability of the approach. Of course, agents can potentially have a huge number of input states, and thus the decision tree might become large, making the approach less efficient. Typically however, only a limited number of input states are relevant for norm violation, thus limiting the size of the decision tree (by pruning the tree). Furthermore, the size of the tree can also be reduced by pruning parts of the tree that always result in norm violation or no violation of the norm.

Within the domain of normative organizations, a lot of work has been done. In Searle (1995) a distinction is made between constitutive norms and regulative norms. Hereby constitutive norms create new states of affairs and regulative norms govern activities, for example expressing obligations and permission for performing actions. The approach presented in this paper addresses both types of norms. In e.g. Rubino et al (2007), Artikis (2003), and Esteva et al (2001) approaches for representing norm-governed multi-agent systems are introduced. This paper takes such specified norms as input and enables avoidance of violation of these norms. More specific for norm violation, in Beolla and Van der Torre (2008) procedural norms are discussed that can for example be used to motivate agents that play a role in recognizing violations or applying sanctions. In Vazquez-Salceda (2004) a formalism is introduced to detect, sanction and repair violations of norms. The approach presented in our paper is however meant to *avoid* this violation not to punish violations that have already taken place. This phenomenon can also be seen in society whereby e.g. probation officers track potential recidivists and try to avoid criminal behavior from occurring (Pease, 2002). Of course, the sanctions and repairs can still be very useful in case norm violation is unavoidable.

REFERENCES

Artikis, A., Executable Specification of Open Norm-Governed Computational Systems, PhD thesis, Dept. of E. and Electr. Eng., Imperial Coll. London, Nov. 2003.

Boella, G., and Torre, L. van der, Substantive and Procedural Norms in Normative Multiagent Systems. Journal of Applied Logic, volume 6, issue 2, pp.152-171, 2008.

Castelfranchi, C., Engineering Social Order. In: Omnici, A., Tolksdorf, R., Zambonelli, F. (eds.), Proc. of the First Int. Workshop on Engineering Societies in the Agent World, LNAI 971, pp. 1-18, Springer-Verlag, 2000.

Cohen, L.E. and Felson, M. (1979) Social change and crime rate trends: a routine activity approach. American Sociological Review, vol. 44, pp.588-608.

Esteva, M., Padget, J., and Sierra, C., Formalizing a language for institutions and norms. In Meyer, J.J.Ch. and Tambe, M. (eds.), Intelligent Agents VIII, LNAI 2333, pp 348–366, Springer-Verlag 2001.

Ferber, J. and Gutknecht, O., A meta-model for the analysis and design of organisations in multi-agent systems. In: Proceedings of the Third Int. Conf. on Multi-Agent Systems (ICMAS'98), IEEE Computer Society Press, pp. 128-135.

Hannoun, M., Boissier, O., Sichman, J.S., and Sayettat, C., MOISE: An organizational model for multi-agent systems. In: Monard, M. C. and Sichman, J. S. (eds.), Advances in AI, LNAI, vol. 1952, Springer, Berlin, 2000, pp. 152-161.

Pease, K., Crime Reduction. In: Maguire, M., Morgan, R., and Reiner, R. (eds), The Oxford Handbook of Criminology. New York: Oxford U. Press Inc, 2002.

Quinlan, J.R., Discovering rules by induction from large collections of examples. In: Michie, D. (Ed.), Expert systems in the micro electronic age. Edinburgh U. Pr. 1979.

Rao, A.S. and Georgeff, M.P., Modeling Rational Agents within a BDI-Architecture. In Proc. of the 2nd Int. Conf. on Principles of Knowledge Repr. and Reasoning, pp. 473-484, 1991.

Rubino, R., Omnincini, A., and Denti, E., Computational Institutions for Modeling Norm-Regulated MAS: An Approach Based on Coordination Artifacts. In: Bossier, O et al (eds.) ANIREM and OOOP 2005, LNAI 3913, pp. 127 – 141, Springer-Verlag 2007.

Searle, J.R., The construction of social reality. New York: The Free Press, 1995.

Vazquez-Salceda, J., Aldewereld, H., Dignum, F., Implementing Norms in Multiagent Systems. In: Lindemann et al. (eds.), MATES 2004, LNAI 3187, pp. 313-327, Springer, 2004.

AUTHOR BIOGRAPHIES

CHARLOTTE GERRITSEN is a PhD-student at the VU University in Amsterdam. After having finished her masters in Law and Criminology, Charlotte Gerritsen started her PhD-project at the department of Artificial Intelligence at the VU University in Amsterdam. Her main research interests are (multi-)agent systems, cognitive modeling and simulations. Her research focuses on the domain of crime.

MARK HOOGENDOORN is an assistant professor at the VU University Amsterdam, Department of Artificial Intelligence. Before starting as an assistant professor he has been a visiting researcher at the University of Minnesota, Department of Computer Science and Engineering. He obtained his PhD degree from the VU University in 2007. In his PhD research he focused on organizational change within multi-agent systems, applying his research in projects in various domains, including incident management, logistics, and the naval domain. His current research interests include multi-agent systems, cognitive modeling, and ambient intelligence.