An Attacker Model for Normative Multi-agent Systems

Guido Boella[1] and Leendert van der Torre[2]

[1] Dipartimento di Informatica, Università degli Studi di Torino, Italy
guido@di.unito.it
[2] Computer Science and Communication (CSC), University of Luxembourg
leendert@vandertorre.com

Abstract. In this paper we introduce a formal attacker model for normative multi-agent systems. In this context, an attacker is an agent trying to profit from norm violation, for example because the violation is not detected, it is not being sanctioned, or the sanction is less than the profit of violation. To deliberate about norm violations, an attacker has a self model and a model of the normative multi-agent system, which in our case have the same structure. Moreover, we assume that an attacker violates a norm only when it profits from it, and the attacker therefore plays a violation game with the system. On a variety of examples, we show also how our model of violation games based also on agent abilities or power extends our earlier model based on motivations only.

1 Introduction

To study the security of a normative multi-agent system [3], we have to build a model of an attacker or adversary. Though explicit attacker models are well known and studied in cryptography and computer security, it seems that the concept has not been developed thus far for normative multi-agent systems.

In this context, an attacker is an agent trying to profit from norm violation, for example because the violation is not detected, it is not being sanctioned, or the sanction is less than the profit of violation [1]. Attacker models extend our earlier violation games studying fraud and deception [2], because these games consider only the agents' decision variables directly controlled by the agents, without modelling the preconditions of the decisions or their consequences.

In this paper we show that if we consider also the agent's ability or power to change the state of the world via conditional effects of decisions, we can model new interesting violation games. In particular, the attacker can exploit indirect conflicts between effects of decision variables, and incompatible decisions.

The layout of this paper is as follows. We first introduce our earlier model by describing the range of violation games that can be played in it, together with various violation games involving an attacker that are not covered by the previous model. Then we introduce our new formal attacker model, and finally we illustrate the model by formalizing some of the examples.

H.-D. Burkhard et al. (Eds.): CEEMAS 2007, LNAI 4696, pp. 42–51, 2007.

2 Violation Games of Attackers of Normative Systems

Normally an agent fulfills its obligations, because otherwise its behavior counts as a violation that is sanctioned, and the agent dislikes sanctions [1]. Moreover, it may dislike violations regardless of sanctions, and it may act according to the norm regardless whether its behavior counts as a violation. In this section we discuss four main exceptions to this normal behavior, the former two taken from [2] are related to motivations only, the latter two are dealing also with abilities or powers of agents.

First, an agent may violate an obligation when the violation is preferred to the sanction, like people who like to speed and do not care about the speeding tickets. In such cases the system may increase its sanctions. An instance occurs when the agent is sanctioned and does not care about additional sanctions. In this case the additional sanction may not be high enough, but often the first sanction was too high. An example is the famous Beccaria's argument against death penalty: it makes sanctions associated to other crimes irrelevant.

Moreover, the agent may have conflicting desires or obligations which it considers more important. First, obligations may conflict with each other. E.g., if the normative system for web access is incoherent it may forbid and oblige at the same time to access a robot.txt file. Second, obligations may conflict with the agent's private preferences, since they are adopted as goals by agents who respect them. The agent can desire exactly the opposite it is obliged to, e.g., in a file sharing system a record company prefers not to share its music files. In the case of conflict with goals, the matter becomes more complex: besides deciding whether the obligation is preferred to the goal, the agent must consider whether it should change its mind. Different types of agents can have different attitudes towards goal reconsideration. Third, if an agent does not fulfill an obligation, then there is a conflict between the sanction it is subject to and its desires and goals, since by definition the sanction must not be preferred. E.g., the sanction can be the inability to access a resource or the loss of a monetary deposit.

Second, the agent may think that its behavior will not be counted as a violation, or that it will not be sanctioned. This may be the case in the normal situation when the agents working for the normative system are lazy, because the sanction has a cost for the system, but more likely it is due to an action of the agent. This action may abort the goal of the normative system to count the agent's behavior as a violation or to sanction it, as in the case of bribing, or it may trigger a conflict for the normative system. An agent can use recursive modelling [6] to exploit desires and goals of the normative system thus modifying its motivations and inducing it not to sanction. In particular, the agent can work on the conditional desires and goals [5] of the system by triggering some of them which conflict with the goal of sanctioning the agent. We distinguish two situations. In the first case, the normative system has no advantage from the action executed by the agent. E.g., in some countries members of parliament cannot be sanctioned, so an agent can become an MP in order not to be punished. In the second case, the normative system gains an advantage from the action of the agent. A particular case of this behavior is bribing: some agents working for the

normative system have a disposition to be bribed, i.e., they have a goal towards not sanctioning the agent that can be triggered by a payment by part of the agent; if the payment to the system is smaller than the cost of disk space, then the agent profits by bribing agents working for the system.

Third, no motivation can lead the agent to fulfill an obligation if it cannot achieve it: e.g., in a file sharing system the agent may have already reached its quota of disk space, so it has no space left to put at disposal of the community it belongs to. Moreover, the difference with the examples in [2] is that with abilities there is not necessarily an explicit conflict in the obligations posed by the normative system: the agent may not have enough resources to fulfill all the obligations, or the actions for fulfilling the obligations have incompatible effects. Conflicts may arise indirectly also with actions the agent desires to perform, for example, it cannot use the disk space for installing software it needs.

Fourth, the normative system can be unable to count the behavior as a violation or sanction it. This behavior is caused by an action of the agent, either by blocking the sanction directly, or creating a conflict with other obligations. Concerning influencing the normative system, the explicit modelling of the system's abilities or power allows more ways for the agent to violate obligations while avoiding the sanction. The recognition of a violation and the sanction may require some applicability conditions to be achieved. Hence the agent can manipulate these conditions: it can make it impossible for the system to prosecute it or to perform the sanction. E.g., an access control system is not able anymore to block the agent's connections, since it has changed its IP address. Alternatively, the agent can make it more difficult, and hence more costly, for the system to execute the prosecution process and the sanction. E.g., some of the agent's actions can in fact 'trigger' some side effect of the action of sanctioning. The agent can use proxy servers to connect to some access control system, so that it is more difficult to block the agent's connections. This additional cost may make it not worthwhile to enforce the respect of the obligation.

Moreover, another way of changing the system's behavior is to trigger some of its other goals, so that the system is led to plan a solution for achieving also those goals. Which goal is useful to trigger depends on which action the system is compelled to execute to achieve the new goal, if the system prefers to achieve that goal with respect to sanctioning. For example, the agent could launch a denial of service attack against the system. Since for the system it is more important to stop the attack than to apply the sanction and it cannot do both actions, the system drops its decision to sanction the agent. A very particular case of this situation is due to the fact that, in our model, sanctions are modelled as conditional goals which are triggered by a violation. So, paradoxically, the agent can trigger the system's action by violating some other obligation whose sanction makes the normative system want not to apply further sanctions. If this new sanction is less harsh then the first one and the system is compelled to punish only the second obligation first, e.g., the goal to apply the first sanction prevails to the goal of applying the second one, then the agent can even benefit from violating two obligations while being sanctioned for only one.

3 Attacker Model for Normative Multiagent Systems

To deliberate about norm violations, an attacker model consists of:

A self model of the attacker, distinguishing, for example, between norm internalizing agents, respectful agents and selfish agents. In this paper we use a Belief-Desire-Goal model to represent the attacker's mental attitudes.

A model of the normative multi-agent system, where for efficiency reasons we assume that the attacker describes the normative multi-agent system with the same structures used to describe its self-model. The beliefs, desires and goals of the system reflect these mental attitudes of agents working for it, such as legislators, judges, and policemen.

A model of interaction, where the agent plays a so-called violation game with the system, based on recursive modeling [6].

In the definition of multiagent system, not necessarily all variables are assigned to an agent, and we introduce effect rules. The description of the world contains – besides decision variables whose truth value is determined directly by an agent – also *parameters* whose truth value can only be determined indirectly. The distinction between decision variables and parameters is a fundamental principle in all decision theories or decision logics [4,7]. To distinguish the model developed in [2] from the one developed in this paper, we call the model in this paper our second normative multi-agent system.

Definition 1 (Agent description). *The tuple* $\langle A, X, D, G, AD, E, MD, \geq \rangle$ *is our second multiagent system* MAS_2*, where*

- *the agents* A*, variables* X*, desires* D *and goals* G *are four finite disjoint sets. We write* $M = D \cup G$ *for the motivations defined as the union of the desires and goals.*
- *an agent description* $AD : A \to 2^{X \cup D \cup G}$ *is a complete function that maps each agent to sets of decision variables, desires and goals. For each agent* $a \in A$*, we write* X_a *for* $X \cap AD(a)$*,* D_a *for* $D \cap AD(a)$*,* G_a *for* $G \cap AD(a)$*. We write parameters* $P = X \setminus \cup_{a \in A} X_a$*.*
- *the set of literals built from* X*, written as* $L(X)$*, is* $X \cup \{\neg x \mid x \in X\}$*, and the set of rules built from* X*, written as* $R(X) = 2^{L(X)} \times X$*, is the set of pairs of a set of literals built from* X *and a literal built from* X*, written as* $\{l_1, \ldots, l_n\} \to l$*. We also write* $l_1 \wedge \ldots \wedge l_n \to l$ *and when* $n = 0$ *we write* $\top \to l$*. Moreover, for* $x \in X$ *we write* $\sim x$ *for* $\neg x$ *and* $\sim(\neg x)$ *for* x*.*
- *the set of effects* $E \subseteq R(X)$ *is a set of rules built from* X*.*
- *the motivational description* $MD : M \to R(X)$ *is a complete function from the sets of desires and goals to the set of rules built from* X*. For a set of motivations* $S \subseteq M$*, we write* $MD(S) = \{MD(s) \mid s \in S\}$*.*
- *a priority relation* $\geq$*:* $A \to 2^M \times 2^M$ *is a function from agents to a transitive and reflexive relation on the powerset of the motivations containing at least the subset relation. We write* $\geq_a$ *for* $\geq (a)$*.*

We model the attacker and the normative multiagent system as two agents **a** and **n**, such that we can describe the interaction between the attacker and the system as a game between two agents. In the definition of normative multiagent system, we add the fact that violations can be mapped on parameters as well as decision variables of system **n**, which formalizes the property that in some circumstances system **n** may not be able to count behavior as violation. Moreover, we change the definition of goal distribution such that agents can also be responsible for parameters.

Definition 2 (Norm description). *Let $\langle A, X, D, G, AD, E, MD, \geq\rangle$ be our second multiagent system. The tuple $\langle A, X, D, G, AD, E, MD, \geq, \mathbf{n}, N, V, GD\rangle$ is our second normative multiagent system $NMAS_2$, where:*

- *the normative system $\mathbf{n} \in A$ is an agent.*
- *the norms $N = \{n_1, \ldots, n_m\}$ is a set disjoint from A, X, D, and G.*
- *the norm description $V : N \times A \rightarrow X_{\mathbf{n}} \cup P$ is a complete function from the norms to the decision variables of the normative agent together with the parameters: we write $V(n, a)$ for the decision variable which represents that there is a violation of norm n by agent $a \in A$.*
- *the goal distribution $GD : A \rightarrow 2^{G_{\mathbf{n}}}$ is a function from the agents to the powerset of the goals of the normative agent, where $GD(a) \subseteq G_{\mathbf{n}}$ represents the goals of agent **n** the agent a is responsible for, such that if we have $L \rightarrow l \in MD(GD(a))$, then $l \in L(X_{\mathbf{a}} \cup P)$.*

4 Obligations in the Normative Multi-agent System

We introduce a logic of rules *out* for the desires and goals of the agents, and a second logic of rules for the effect rules, called *outE*. Both take the transitive closure of a set of rules, called reusable input/output logic in [8], but the latter also includes the input in the output.

Definition 3 (Out). *Let $MAS_2 = \langle A, X, D, G, AD, E, MD, \geq\rangle$ be our second multiagent system. Moreover, let:*

- *$out(D, S)$ be the closure of $S \subseteq L(X)$ under the rules $D \subseteq R(X)$:*
  - *$out^0(D, S) = \emptyset$*
  - *$out^{i+1}(D, S) = out^i(D, S) \cup \{l \mid L \rightarrow l \in D, L \subseteq out^i(D, S) \cup S\}$ for $i \geq 0$*
  - *$out(D, S) = \cup_0^\infty out(D, S)$*

  Moreover, following notational conventions in input/output logics, we write $a \rightarrow x \in out(R)$ for $x \in out(R, \{a\})$.
- *$outE(E, S)$ be the closure of $S \subseteq L(X)$ under the effect rules E:*
  - *$outE^0(E, S) = S$*
  - *$outE^{i+1}(E, S) = outE^i(E, S) \cup \{l \mid L \rightarrow l \in E, L \subseteq outE^i(E, S)\}$ for $i \geq 0$*
  - *$outE(E, S) = \cup_0^\infty outE(E, S)$*

  Moreover, we write $a \rightarrow x \in outE(R)$ for $x \in outE(R, \{a\})$.

Example 1. Consider the multiagent system $\langle A, X, D, G, AD, E, MD, \geq \rangle$ where $A = \{\mathbf{a}, \mathbf{n}\}$, $X_{\mathbf{a}} = \{x\}$, $X_{\mathbf{n}} = \{y\}$. Agent $\mathbf{a}$ desires and wants unconditionally to decide to do x, but if agent $\mathbf{n}$ decides y, then it wants the opposite $\neg x$: $MD(D_{\mathbf{a}}) = MD(G_{\mathbf{a}}) = \{\top \rightarrow x, y \rightarrow \neg x\}$. The second rule is preferred over the first one, the ordering $\geq_{\mathbf{a}}$ is $\{\top \rightarrow x, y \rightarrow \neg x\} > \{y \rightarrow \neg x\} > \{\top \rightarrow x\} > \emptyset$. System $\mathbf{n}$ desires and wants to do y: $MD(D_{\mathbf{n}}) = MD(G_{\mathbf{n}}) = \{\top \rightarrow y\}$. Moreover, assume $P = \{p, q\}$ and $E = \{\top \rightarrow p, x \rightarrow q, y \rightarrow \neg q\}$. The output is given in Figure 1. The unconditional effect rule $\top \rightarrow p$ represents what is true in the initial state. $outE(E, \{x\}) = \{x, p, q\}$, while $outE(E, \{y\}) = \{x, p, \neg q\}$. The remainder of this figure is explained in the following section.

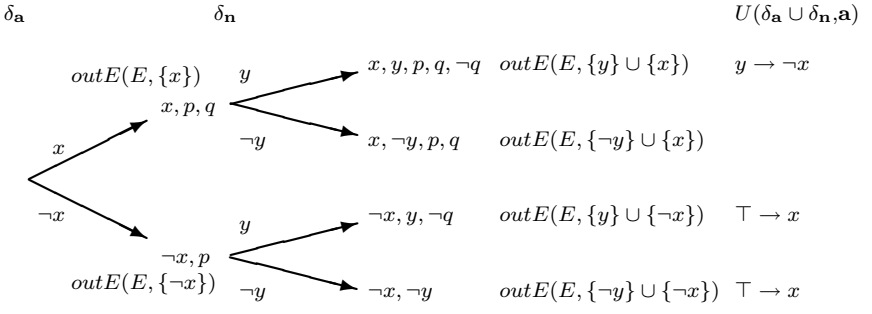

Fig. 1. Violation game tree

The obligations, count-as-violations as well as sanctions can be parameters, which can only be achieved indirectly by the agents' decisions. For count-as-violations and sanctions, we extend the definition of obligation with the additional clause that the normative agent has at least one way to apply the sanction. To formalize this clause, we already have to define what decisions are (which in [2] we could delay until the section on behavior).

Definition 4 (Decisions). *The set of decisions Δ is the set of subsets $\delta \subseteq L(X \setminus P)$ such that their closure under effect rules $outE(E, \delta)$ does not contain a variable and its negation. For an agent $a \in A$ and a decision $\delta \in \Delta$ we write δ_a for $\delta \cap L(X_a)$.*

Given decisions we define the ability of an agent to make true a propositional variable in a certain context by means of a decision:

Definition 5 (Ability). *Agent $a \in A$ is able to achieve $p \in L(X)$ in context $Y \subseteq L(X)$, written as $able(a, p, Y)$, if and only if there is a decision δ_a such that $p \in outE(E, Y \cup \delta_a)$.*

In the following section we show that for our recursive games the new clauses do not imply that the normative agent can always count behavior as a violation and sanction it. The reason is that some decisions of the normative multi-agent system can be blocked due to decisions of agent $\mathbf{a}$. E.g., if agent $\mathbf{a}$ sees to it that

a parameter p is true, then all decisions of system $\mathbf{n}$ are blocked that would see to $\neg p$, because the effects of decision must be consistent.

Definition 6 (Obligation). *Let*

$$NMAS_2 = \langle A, X, D, G, AD, E, MD, \geq, \mathbf{n}, N, V, GD \rangle$$

be our second normative multiagent system. In $NMAS_2$ agent $\mathbf{a} \in A$ is obliged to decide to do $x \in L(X_\mathbf{a} \cup P)$ with sanction $s \in L(X_\mathbf{n} \cup P)$ if $Y \subseteq L(X_\mathbf{a} \cup P)$, written as $NMAS_2 \models O_\mathbf{an}(x, s|Y)$, if and only if $\exists n \in N$ such that:

1. $Y \rightarrow x \in out(D_\mathbf{n}) \cap out(GD(\mathbf{a}))$*: if Y then system $\mathbf{n}$ desires and has as a goal that x, and this goal has been distributed to agent $\mathbf{a}$.*
2. $Y \cup \{\sim x\} \rightarrow V(n, \mathbf{a}) \in out(D_\mathbf{n}) \cap out(G_\mathbf{n})$*: if Y and $\sim x$ is done by agent $\mathbf{a}$, then system $\mathbf{n}$ has the goal and the desire $V(n, \mathbf{a})$: to recognize it as a violation done by agent $\mathbf{a}$.*
3. $\top \rightarrow \neg V(n, \mathbf{a}) \in out(D_\mathbf{n})$*: system $\mathbf{n}$ desires that there are no violations.*
4. $Y \cup \{V(n, \mathbf{a})\} \rightarrow s \in out(D_\mathbf{n}) \cap out(G_\mathbf{n})$*: if Y and system $\mathbf{n}$ decides $V(n, \mathbf{a})$ then system $\mathbf{n}$ desires and has as a goal that it sanctions agent $\mathbf{a}$.*
5. $Y \rightarrow \sim s \in out(D_\mathbf{n})$*: system $\mathbf{n}$ desires not to sanction, $\sim s$. This desire of the normative system expresses that it only sanctions in case of a violation.*
6. $Y \rightarrow \sim s \in out(D_\mathbf{a})$*: if Y, then agent $\mathbf{a}$ desires $\sim s$, which expresses that it does not like to be sanctioned.*
7. $able(\mathbf{n}, V(n, \mathbf{a}), Y \cup \{\sim x\})$*: system $\mathbf{n}$ is able to achieve $V(n, \mathbf{a})$ in context Y and $\sim x$;*
8. $able(\mathbf{n}, s, Y \cup \{V(n, \mathbf{a}), \sim x\})$*: system $\mathbf{n}$ is able to achieve s in context Y, $V(n, \mathbf{a})$ and $\sim x$.*

An obligation $O_\mathbf{an}(x, s|Y)$ is an ought-to-do obligation when $x \subseteq L(X_\mathbf{a})$, and an ought-to-be obligation otherwise.

The following example illustrates that the agent does not always know how to fulfill ought-to-be obligations.

Example 2. Consider $\langle A, X, D, G, AD, E, MD, \geq, \mathbf{n}, N, V, GD \rangle$ with $A = \{\mathbf{a}, \mathbf{n}\}$, $X_\mathbf{a} = \{b\}$, $X_\mathbf{n} = \{V(n, \mathbf{a}), m\}$, $P = \{p, x\}$, $N = \{n\}$, $MD(GD(\mathbf{a})) = \{\top \rightarrow x\}$. Agent $\mathbf{a}$ desires $\neg s$: $MD(D_\mathbf{a}) = \{\top \rightarrow \neg s\}$. Agent $\mathbf{n}$ desires and goals are: $MD(D_\mathbf{n}) = \{\top \rightarrow x, \neg x \rightarrow V(n, \mathbf{a}), V(n, \mathbf{a}) \rightarrow s, \top \rightarrow \neg V(n, \mathbf{a}), \top \rightarrow \neg s\}$ and $MD(G_\mathbf{n}) = \{\top \rightarrow x, \neg x \rightarrow V(n, \mathbf{a}), V(n, \mathbf{a}) \rightarrow s, \top \rightarrow \neg V(n, \mathbf{a}), \top \rightarrow \neg s\}$.

Assume $E = \{\top \rightarrow p, b \rightarrow x, p \rightarrow \neg x\}$. In this situation, agent $\mathbf{a}$ cannot fulfill the obligation, because there does not exist a decision $\delta = \delta_\mathbf{a} \cup \delta_\mathbf{n}$ such that $x \in outE(E, \delta)$. The parameter x could be achieved by means of action b since $b \rightarrow x \in E$. But in the initial state the parameter p is true ($\top \rightarrow p \in E$) and $p \rightarrow \neg x \in E$: hence, $\{b\}$ is not a decision of agent $\mathbf{a}$, otherwise we have $\{x, \neg x\} \in outE(E, \delta')$.

5 Formalization of Violation Games of Attacker

The basic picture is visualized in Figure 1 and reflects the deliberation of agent **a** in various stages. This figure should be read as follow. Agent **a** is the decision maker: it is making a decision $\delta_{\mathbf{a}}$, and it is considering the effects of the fulfilment or the violation of the obligations it is subject to. To evaluate a decision $\delta_{\mathbf{a}}$ according to its desires and goals ($D_{\mathbf{a}}$ and $G_{\mathbf{a}}$), it must consider not only its actions, but also the reaction of system **n**: **n** is the normative system, which may recognize and sanction violations. Agent **a** recursively models system **n**'s decision $\delta_{\mathbf{n}}$ (that system **n** takes according to agent **a**'s point of view), typically whether it counts the decision $\delta_{\mathbf{a}}$ as a violation and whether it sanctions agent **a** or not, and then bases its decision on it. Now, to find out which decision system **n** will make, agent **a** has a *profile* of system **n**: it has a representation of system **n**'s motivational state. When agent **a** makes its decision and predict system **n**'s decision, we assume in this paper that it believes that system **n** is aware of it.

Definition 7 (Recursive modelling). *Let*

$$NMAS_2 = \langle A, X, D, G, AD, E, MD, \geq, \mathbf{n}, N, V, GD \rangle$$

be a normative multiagent system.

- *Let the unfulfilled motivations of decision δ for agent $a \in A$ be the set of motivations whose body is part of the closure of the decision under the effect rules but whose head is not.*
  $U(\delta, a) = \{m \in M \cap MD(a) \mid MD(m) = l_1 \wedge \ldots \wedge l_n \rightarrow l, \{l_1, \ldots, l_n\} \subseteq outE(E, \delta)$ *and* $l \notin outE(E, \delta)\}$
- *A decision δ (where $\delta = \delta_{\mathbf{a}} \cup \delta_{\mathbf{n}}$) is* optimal *for agent* **n** *if and only if there is no decision $\delta'_{\mathbf{n}}$ such that $U(\delta, \mathbf{n}) >_{\mathbf{n}} U(\delta_{\mathbf{a}} \cup \delta'_{\mathbf{n}}, \mathbf{n})$. A decision δ is optimal for agent* **a** *and agent* **n** *if and only if it is optimal for agent* **n** *and there is no decision $\delta'_{\mathbf{a}}$ such that for all decisions $\delta' = \delta'_{\mathbf{a}} \cup \delta'_{\mathbf{n}}$ and $\delta_{\mathbf{a}} \cup \delta''_{\mathbf{n}}$ optimal for agent* **n** *we have that $U(\delta', \mathbf{a}) >_{\mathbf{a}} U(\delta_{\mathbf{a}} \cup \delta''_{\mathbf{n}}, \mathbf{a})$.*

In Example 3, the desire of agent **a** (that p is true) conflicts in an indirect way with x, the normative goal of $O_{\mathbf{an}}(x, s \mid \top)$. The effect rules E represent the incompatibility of the effects of the two decision variables x (which has effect $\neg p$) and b (which achieves p). $\{x, b\}$ is not a decision of agent **a** since $outE(E, \{x, b\}) = \{x, b, p, \neg p\}$ is not consistent.

The choice between x and b is taken by comparing the results of recursive modelling: if x then $outE(E, \delta_{\mathbf{a}} \cup \delta_{\mathbf{n}}) = \{x, \neg b, \neg p, \neg V(n, \mathbf{a}), \neg s\}$ and if b then $outE(E, \delta_{\mathbf{a}} \cup \delta_{\mathbf{n}}) = \{b, \neg x, V(n, \mathbf{a}), s\}$. Since agent **a** prefers not being sanctioned with respect to leaving p unfulfilled, it chooses x. The priority order on the motivations is implicitly represented by the numerical index of the rules. A higher number represents a higher priority.

Example 3. $O_{\mathbf{an}}(x, s \mid \top)$

		P	p
		E	$x \rightarrow \neg p, b \rightarrow p$
Agent **a**		Agent **n**	
$X_{\mathbf{a}}$	x, b	$X_{\mathbf{n}}$	$V(n, \mathbf{a}), s$
$D_{\mathbf{a}}$	$\top \rightarrow^2 \neg s, \top \rightarrow^1 p$	$D_{\mathbf{n}}$	$\top \rightarrow^5 x, \neg x \rightarrow^4 V(n, \mathbf{a}), V(n, \mathbf{a}) \rightarrow^3 s,$ $\top \rightarrow^2 \neg V(n, \mathbf{a}), \top \rightarrow^1 \neg s$
$G_{\mathbf{a}}$		$G_{\mathbf{n}}$	$\top \rightarrow^5 x, \neg x \rightarrow^4 V(n, \mathbf{a}), V(n, \mathbf{a}) \rightarrow^3 s$
$\delta_{\mathbf{a}}$	$x, \neg b$	$\delta_{\mathbf{n}}$	$\neg V(n, \mathbf{a}), \neg s$
$U_{\mathbf{a}}$	$\top \rightarrow^1 p$	$U_{\mathbf{n}}$	
	$outE(E, \delta)$	$x, \neg b, \neg p, \neg V(n, \mathbf{a}), \neg s$	

In the next example, agent **a** has to choose between two obligations which, even if they are not explicitly conflicting, it cannot respect at the same time, since fulfilling the first one (the *ought-to-do* obligation $O_{\mathbf{an}}(x, s \mid \top)$) makes it impossible to do anything for the second one (the *ought-to-be* obligation $O_{\mathbf{an}}(p, s' \mid \top)$).

Example 4. $O_{\mathbf{an}}(x, s \mid \top), O_{\mathbf{an}}(p, s' \mid \top)$

		P	p
		E	$x \rightarrow \neg p, y \rightarrow p$
Agent **a**		Agent **n**	
$X_{\mathbf{a}}$	x, y	$X_{\mathbf{n}}$	$V(n, \mathbf{a}), s, V(n', \mathbf{a}), s'$
$D_{\mathbf{a}}$	$\top \rightarrow^2 \neg s, \top \rightarrow^1 \neg s'$	$D_{\mathbf{n}}$	$\top \rightarrow^{10} x, \neg x \rightarrow^9 V(n, \mathbf{a}), V(n, \mathbf{a}) \rightarrow^8 s,$ $\top \rightarrow^4 \neg V(n, \mathbf{a}), \top \rightarrow^3 \neg s,$ $\top \rightarrow^7 p, \neg p \rightarrow^6 V(n', \mathbf{a}), V(n', \mathbf{a}) \rightarrow^5 s',$ $\top \rightarrow^2 \neg V(n', \mathbf{a}), \top \rightarrow^1 \neg s'$
$G_{\mathbf{a}}$		$G_{\mathbf{n}}$	$\top \rightarrow^{10} x, \neg x \rightarrow^9 V(n, \mathbf{a}), V(n, \mathbf{a}) \rightarrow^8 s,$ $\top \rightarrow^7 p, \neg p \rightarrow^6 V(n', \mathbf{a}), V(n', \mathbf{a}) \rightarrow^5 s'$
$\delta_{\mathbf{a}}$	$x, \neg y$	$\delta_{\mathbf{n}}$	$\neg V(n, \mathbf{a}), s, V(n', \mathbf{a}), s'$
$U_{\mathbf{a}}$	$\top \rightarrow^1 \neg s'$	$U_{\mathbf{n}}$	$\top \rightarrow^7 p, \top \rightarrow^2 \neg V(n', \mathbf{a}), \top \rightarrow^1 \neg s'$
	$outE(E, \delta)$	$x, \neg y, \neg p, \neg V(n, \mathbf{a}), \neg s, V(n', \mathbf{a}), s'$	

In the next example, we again model the sanction s as a parameter which is made true by a decision variable $m \in X_{\mathbf{n}}$. However, this time agent **a** does not directly make the sanction impossible. Rather, it triggers some goals of agent **n**. We examine how agent **a** exploits the recursive modelling to influence the behavior of the normative agent. Besides the usual goals and desires described by the obligation to do x, here we assume that, in a situation where p is true, agent **n** has the goal to make the decision variable $r \in X_{\mathbf{n}}$ true. So, given $p \in P$, it would like to choose decision $\delta_{\mathbf{n}} = \{V(n, \mathbf{a}), m, r\}$: but the two decision variables m and r are incompatible. Since the conditional desire $p \rightarrow r \in out(D_{\mathbf{n}})$ is preferred to $V(n, \mathbf{a}) \rightarrow s \in out(D_{\mathbf{n}})$, agent **n** recognizes the violation ($V(n, \mathbf{a})$) but it does not sanction agent **a**.

Example 5. $O_{\mathbf{an}}(x, s \mid \top)$

P	p, s
E	$b \to p, m \to s, r \to \neg s$
Agent **a**	Agent **n**
$X_{\mathbf{a}}$ x, b	$X_{\mathbf{n}}$ $V(n, \mathbf{a}), m, r$
$D_{\mathbf{a}}$ $\top \to^3 \neg s, \top \to^2 \neg x, \top \to^1 \neg b$	$D_{\mathbf{n}}$ $p \to^6 r, \top \to^5 x, \neg x \to^4 V(n, \mathbf{a}),$ $V(n, \mathbf{a}) \to^3 s,$ $\top \to^2 \neg V(n, \mathbf{a}), \top \to^1 \neg s$
$G_{\mathbf{a}}$	$G_{\mathbf{n}}$ $p \to^6 r, \top \to^5 x, \neg x \to^4 V(n, \mathbf{a}),$ $V(n, \mathbf{a}) \to^3 s$
$\delta_{\mathbf{a}}$ $b, \neg x$	$\delta_{\mathbf{n}}$ $V(n, \mathbf{a}), r, \neg m$
$U_{\mathbf{a}}$ $\top \to^1 \neg b$	$U_{\mathbf{n}}$ $\top \to^5 x, V(n, \mathbf{a}) \to^3 s, \top \to^2 \neg V(n, \mathbf{a})$
$outE(E, \delta)$	$\neg x, p, V(n, \mathbf{a}), b, \neg m$

6 Summary

With the increase of multiagent systems with explicit norms, their security becomes an urgent problem. To deal with a wide range of possible attacks, we need an expressive attacker model. In this paper we considered the abilities or power of agents, and two new kinds of examples, either due to the inability or lack of power of the attacker itself, or due to the inabilities or lack of power of the normative system. A topic of further research is the extension of our model to reason about beliefs and observations of attackers and normative systems [1].

References

1. Boella, G., van der Torre, L.: Fulfilling or violating obligations in normative multiagent systems. In: Procs. of IAT'04, pp. 483–486. IEEE, Los Alamitos (2004)
2. Boella, G., van der Torre, L.: Normative multiagent systems and trust dynamics. In: Falcone, R., Barber, S., Sabater-Mir, J., Singh, M.P. (eds.) Trusting Agents for Trusting Electronic Societies. LNCS (LNAI), vol. 3577, pp. 1–17. Springer, Heidelberg (2005)
3. Boella, G., van der Torre, L., Verhagen, H.: Introduction to normative multiagent systems. Computation and Mathematical Organizational Theory, Special issue on Normative Multiagent Systems 12(2-3), 71–79 (2006)
4. Boutilier, C.: Toward a logic for qualitative decision theory. In: Procs. of KR-94, Bonn, pp. 75–86 (1994)
5. Broersen, J., Dastani, M., Hulstijn, J., van der Torre, L.: Goal generation in the BOID architecture. Cognitive Science Quarterly 2(3-4), 428–447 (2002)
6. Gmytrasiewicz, P.J., Durfee, E.H.: Formalization of recursive modeling. In: Procs. of ICMAS'95, pp. 125–132. AAAI/MIT Press, Cambridge, MA (1995)
7. Lang, J., van der Torre, L., Weydert, E.: Utilitarian desires. Autonomous Agents and Multiagent Systems 5(3), 329–363 (2002)
8. Makinson, D., van der Torre, L.: Input-output logics. Journal of Philosophical Logic 29(4), 383–408 (2000)