Specifying and Enforcing Norms in Artificial Institutions [*] (Short Paper)

Nicoletta Fornara[1]
[1]Università della Svizzera italiana
via G. Buffi 13, 6900 Lugano, Switzerland
nicoletta.fornara@lu.unisi.ch

Marco Colombetti [1,2]
[2]Politecnico di Milano
piazza Leonardo Da Vinci 32, Milano,Italy
marco.colombetti@lu.unisi.ch

ABSTRACT

In this paper we investigate two related aspects of the formalization of open interaction systems: how to specify norms, and how to enforce them by means of sanctions. The problem of specifying the sanctions associated with the violation of norms is crucial in an open system because, given that the compliance of autonomous agents to obligations and prohibitions cannot be taken for granted, norm enforcement is necessary to constrain the possible evolutions of the system, thus obtaining a degree of predictability that makes it rational for agents to interact with the system. In our model, we introduce a construct for the definition of norms in the design of artificial institutions, expressed in terms of roles and event times, which, when certain activating events take place, is transformed into commitments of the agents playing certain roles. Norms also specify different types of sanctions associated with their violation. In the paper, we analyze the concept of sanction in detail and propose a mechanism through which sanctions can be applied.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms

Design, Theory

Keywords

Norms, Sanctions, Commitments, Artificial Institutions, Open Interaction Systems.

1. INTRODUCTION

In our previous works [4, 5] we have presented a metamodel of Artificial Institutions (AI) called *OCeAN* (Ontology, CommitmEnts, Authorizations, Norms), which can be used to specify at a high level and in an unambiguous way *open interaction systems*, where heterogeneous and autonomous agents may interact. In our view open interaction systems are a technological extension of human reality, that is, they are an instrument by which human beings can enrich the type and the frequency of their interactions and overcome geographical distance. In this paper we concentrate mainly on the definition of the constructs necessary for the specification of the normative component of AIs, that is, of obligations, permissions and prohibitions of the interacting agents. The normative component is fundamental because it can be used to specify the expected behavior of the interacting agents, for example by means of flexible protocols [13]. We shall extend our *OCeAN* metamodel by defining a construct for the specification of norms for open systems, whose semantics is expressed by means of *social commitments*, the same concept that we have used to specify the semantics of a library of communicative acts [4]. Commitments, having a well defined life-cycle, will be used at run-time to detect and react to the violation of the corresponding norms.

We present an innovative and detailed analysis of problem of defining a mechanism for enforcing obligations and prohibitions by means of sanctions, that is, a treatment of the actions that have to be performed when a violation occurs, in order to deter agents from misbehaving and to secure and recover the system from an undesirable state. The problem of managing sanctions has been tackled in a few other works: for example, López y López et al. [10] propose to enforce norms using the "enforcement norms" that oblige agents entitled to do so to punish misbehaving agents but does not treat the actions that the misbehaving agents may have to perform to repair to its violation; Vázquez-Salceda et al. [12] present, in the OMNI framework, a method to enforce norms described at a different level of abstraction but do not investigate in detail the mechanism to manage santctions; whereas Grossi et al. in [7] develop a high-level analysis of the problem of enforcing norms. Other interesting proposals introduce norms to regulate the interaction in open systems but, even when the problem of enforcement is considered to be crucial, do not investigate with sufficient depth why an agent ought to comply with norms and what would happen if compliance does not occur. For instance, Esteva et al. [3, 6] propose ISLANDER, where a normative language with sanctions is defined but not discussed in detail, Boella et al. [2] model violations but do not analyze sanctions, and Artikis et al. [1] propose a model where the problem of norm enforcement using sanctions is mentioned but not fully investigated.

[*]Supported by Hasler Foundation n°2204 "Artificial institutions: specification of open distributed interaction systems"

Cite as: Specifying and Enforcing Norms in Artificial Institutions (Short Paper), Nicoletta Fornara and Marco Colombetti, *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.), May, 12-16., 2008, Estoril, Portugal, pp. 1481-1484.

2. SANCTIONS

In this section we briefly discuss the crucial role played by *sanctions* in the specification of an open interaction system. In an artificial system, even if the utility function of the misbehaving agent is not known, sanctions can be mainly devised to deter agents from misbehaving bringing about a loss for them in case of violation, under the assumption that the interacting agents are able to reason on sanctions. Moreover sanctions can be devised to compensate the institution or other damaged agents for their loss, or to contribute to the security of the system, for example by prohibiting misbehaving agents to interact any longer with the system or to specify the acts that have to be performed to recover the system from an undesirable state.

When thinking about sanctions from an operational point of view, and in particular to the set of actions that have to be performed when a violation occurs, it is important to distinguish between two types of actions that differ mainly as far as their actors are concerned. One crucial type of action that deserves to be analyzed in detail, and that is not taken into account in other proposals [10, 12, 6], consists of the actions that the misbehaving agent itself has to perform against a violation, and that are devised as a deterrent and/or a compensation for the violation. When trying to model this type of action it is important to take into account that it is also necessary to check that the compensating actions are performed. Another type is characterized by the actions that certain agents are *authorized* to perform only against violations. In other existing proposals, for instance [10, 12], which do not highlight the notion of authorization (or power [8]), those actions are simply the actions that certain agents are obliged to perform against violations. From our point of view, instead, the obligation to sanction a violation should be distinguished from the authorization to do so.

In some situations, and in particular when the sanction is crucial for the continuation of the interaction, one may want to introduce a norm to oblige the agents entitled to do so to manage the violation. Such norm is similar to the "enforcement norm" proposed in [10]: it has to be activated by a violation and its content has to coincide with the sanctions of the violated obligation or prohibition. This norm may in turn be violated, and it is up to the designer of the system to decide when to stop the potentially infinite chain of violations and sanctions, leaving some violation unpunished. Regarding this aspect, to make it reasonable for certain agents (or for their owner) to interact with an open system, it has to be possible to specify that certain violations will definitely be punished (assuming that there are not software failures). One approach is to specify that the actor of the actions performed as sanctions for those violations is the *interaction-system* itself, that therefore needs to be represented in our model as a "special agent". By "special" we mean that such an agent will not be able to take autonomous decisions, and will only be able to follow the system specifications that are stated before the interaction starts. We call this type of agents *heteronomous* (as opposite to autonomous). Examples of reasonable sanctions that can be inflicted by means of norms in an open artificial system are the decrement of the trust or reputation level of the agent, the revocation of the authorization to perform certain actions or a change of role or, as a final action, the expulsion of the agent from the system.

3. NORMS

Norms are taken as a specification of how a system ought to evolve. In an open system, they are necessary to impose obligations and prohibitions to the interacting agents, in order to make the system's evolution at least partially predictable [11]. In particular, norms can be used to express interaction protocols as exemplified in [4], where the English Auction is specified. Our model of norms is mainly suited for *simulating* the evolution of the state of an open system starting from an initial state and for *monitoring* tasks to check whether the agents' behavior is compliant with the specifications and suitably react to violations. Coherent with other approaches [3, 1, 6, 10, 12], in our view norms have to specify who is affected by them, who is the creditor, what are the actions that should or should not be performed, when a given norm is active, and what are the consequences of violating norms.

In the definition of our model it is crucial to distinguish between the definition of a construct for the specification of norms in the design phase, that will be used by human designers, and the specification of how such a construct will evolve during the run-time phase to make it possible to detect and react to norm violations. In particular we assume that during the run-time of the system the interacting agents cannot create new norms, but can create new commitments, directed to specific agents, by performing suitable communicative acts, for example by making promises or by giving orders. During the phase of specification of the set of norms of a certain AI the designer does not know the actual set of agents that will interact with the system at a given time. In this phase it is therefore necessary to define norms based on the notion of role. Moreover, the time instant at which a norm becomes active is typically not known at design time, being related to the occurrence of certain events; for example, the agent playing the role of the auctioneer in an English auction is obliged to declare the current ask-price after receiving each bid by a participant. Therefore at design phase it is only possible to specify the type of event that, if it happens, will activate the norm. During the system run time such a construct of norm, expressed in terms of roles and times of events, must be transformed into an unambiguous representation of the obligations and prohibitions that every agent has at every state of the interaction.

3.1 The construct of norm

A norm is used to impose a certain behavior on certain agents in the system identified by means of the *debtor* attribute and on the basis of the roles they play in the system.

Another fundamental component of a norm is its *content*, which describes the actions that the debtors have to perform (if the norm expresses an obligation) or not to perform (if the norm expresses a prohibition) within a specified interval of time. In our model *temporal propositions* (for a detailed treatment see [4]), are used to represent the content of commitments and, due to the strict connection between commitments and norms, are also used to represent the content of norms. A temporal proposition binds a *statement* about a state of affairs or about the performance of an action to a specific *interval of time* with a certain *mode* (that can be $\forall$ or $\exists$). Temporal propositions are represented with the following notation:

$TP(statement, [t_{start}, t_{end}], mode, truth\text{-}value)$,

where the *truth-value* could be undefined ($\perp$), true or

false. In particular when the *statement* represents the performance of an action and the *mode* is $\exists$, the norm is an obligation and the debtors of the norms have to perform the action within the interval of time. When the *statement* represents the non-performance of an action and the *mode* is $\forall$ the norm is a prohibition and the debtors of the norms should not perform the action within the interval of time. In particular t_{start} is always equal to the time of occurrence of the event that activates the norm. Regarding the verification of prohibitions, in order to be able to check that an action has not been performed during an interval of time it is necessary to rely on the closure assumption that if an action is not recorded as happened in the system, then it has not happened. A norm becomes active when the *activation event* e_{start} happens. Activation can also depend on some Boolean *conditions*, that have to be true in order that the norm can become active; for instance an auctioneer may be obliged to open a run of an auction at time t_{start} if at least two participants are present. An agent can reason whether to fulfil or not to fulfil a norm on the basis of the sanctions/reward and of whom is the *creditor* of the norm, as proposed also in [9, 10]. For example, an agent with the role of auctioneer may decide to violate a norm imposed by the auction house if it is in conflict with another norm that regulates trade transactions in a certain country. Moreover the creditor of a norm is crucial because, given that it becomes the creditor of the commitments generated by the norm (as described in next section), is the only agent authorized to cancel such commitment [4]. Like for the *debtor* attribute, it is useful to express the creditor of declarative norms by means of their role. In order to enforce norms it is necessary to specify sanctions. More precisely it is necessary to specify what actions have to be performed, when a violation occurs, by the debtors of a norm and by the agent(s) in charge of norm enforcement. These two types of actions, that we respectively call *a-sanctions* (active sanctions) and *p-sanctions* (passive sanctions) are sharply dissimilar, and thus require a different treatment. More specifically, to specify a *a-sanction* means to describe an action that the violator should perform in order to extinguish its violation; therefore, it can be specified through a temporal proposition representing an action. On the contrary, to specify a *p-sanction* means to describe what actions the norm enforcer is authorized to perform in the face of a violation; therefore, it can be specified by representing a suitable set of authorizations.

In our model the construct of norm is characterized by the following attributes (their domains is specified in brackets): *debtor* (role), *creditor* (role), *content* (temporal proposition), e_{start}(event-template), *conditions* (Boolean expression), *a-sanctions* (temporal proposition), *p-sanctions* (authorization).

3.2 Commitments with Sanctions

The transformation of norms defined at design time in commitments at run time is crucial because they are the mechanisms used to detect and react to violations. Moreover given that the activation event of norms may happen more than once in the life of the system, it is possible to distinguish between different activations and, in case, violations of the same norm. Given that our previous treatment of *commitment* [4] does not cover sanctions, in this section we extend it to cover this aspect. The content of commitments is expressed using *temporal propositions*. The *state* of a commitment can change as an effect of the execution of institutional actions or of environmental events. Relevant events for the life cycle of commitments are due to the change of the truth-value of the commitment's content. If the content becomes true an event-driven routine automatically changes the commitment's state to fulfilled, otherwise it becomes violated as described in Figure 1.

In our view an operational model of sanctions has to specify how to detect: (i), that a commitment has been violated (a mechanism already introduced in our model of commitment); (ii), that the debtor of the violated commitment performs the compensating actions; and (iii), that the agents entitled to enforce the norms have managed the violation by performing certain actions. Regarding the necessity to check that the debtor performs the compensating actions one possible solution consists in adding two new attributes, *a-sanctions* and *p-sanctions*, to commitments, and two new states, *extinguished* and *irrecoverable*, to their life-cycle. If the actions indicated in the *a-sanctions* attribute are performed, the truth-value of the related temporal proposition becomes true and an event driven routine automatically changes the state of the violated commitment to *extinguished*, as reported in Figure 1. Analogously, if the debtor does not perform those actions, at the end of the specified time interval the truth-value of the temporal proposition becomes false and the state of the commitment becomes *irrecoverable*. Similarly to what we did for norms, the actions that certain agents are authorized to perform against the violation of the commitment are represented in the *p-sanctions* attribute. Note that whether such actions are or are not performed does not affect the life cycle of the commitment; this depends on the fact that the agent that violated a commitment cannot be held responsible for a possible failure of other agents to actually carry out the actions they are authorized to perform. Finally, for proper management of violation it may be necessary to trace the *source* of a commitment, either deriving it from the activation of a norm or from the performance of a communicative act. Our enriched notion of commitment is therefore represented with the following notation:

$$Comm(state, debtor, creditor, content, \\ a\text{-}sanctions, p\text{-}sanctions, source).$$

In our model we use *ECA-rules* (Event-Condition-Action rules), inspired by Active Database models, to specify that certain *actions* are executed when an event identified by an *event-templates* happens, provided that certain Boolean *conditions* are true. The semantics of ECA rules is given as usual: when an event matching the event template occurs in the system, the variable e is filled with the event instance and the condition is evaluated; if the condition is satisfied, the set of actions are executed and their effects are brought about in the system. The *interaction-system* agent (see Section 2) is the actor of the actions performed by means of ECA-rules, and has to have the necessary authorization in order to perform them. In particular the following two ECA-rules have to be present in every interaction system. One in necessary to transform at run time norms into commitments: when the activation event of the norm happens, the *makePendingComm* institutional action is performed and creates a pending commitment for each agent playing one of the roles specified in the *debtors* attribute of the norm:

on e_{start} **if** $norm.conditions$ **then**
do foreach $agent \mid agent.role$ **in** $norm.debtors$

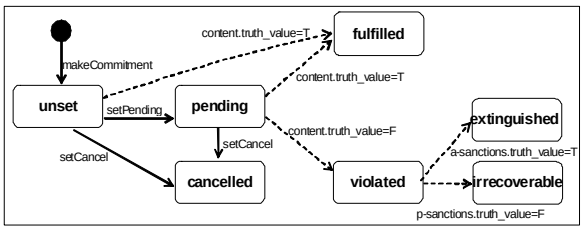

Figure 1: The life-cycle of commitments.

do $makePendingComm(agent, norm.creditor, norm.content$
$norm.a\text{-}sanctions, norm.p\text{-}sanctions, norm\text{-}ref)$

The other is necessary to give the authorizations expressed in the *p-sanctions* attributes to the relevant agents when a commitment is violated:

on e: $AttributeChange(comm.state, violated)$
if $true$ **then do foreach** $auth$ **in** $comm.p\text{-}sanctions$
do $createAuth(auth.role, auth.iaction)$

The *createAuth(role,iaction)* institutional action creates the authorization for the agents playing a certain role to perform a certain institutional action. We assume that the *interaction-system* (the actor of ECA-rules) is always authorized to create new authorizations. A similar ECA-rule has also to be defined to remove such authorizations once *iaction* has been performed.

4. EXAMPLE

An interesting example that highlights the importance of a clear distinction between permission and authorization, which becomes relevant when more than one institution is used to specify the interaction system, is the specification of the Dutch Auction. One of its norms obliges the auctioneer to declare a new ask-price (within λ seconds) lowering the previous one by a certain amount κ, on condition that δ seconds have elapsed from the last declaration of the ask-price without any acceptance act from the participants. If the auctioneer violates this norm the interaction-system is authorized to declare the ask-price and to lower the auctioneer's public reputation level, while the auctioneer has to pay a fine (within h seconds) to extinguish its violation. Such a norm can be expressed in the following way:

$debtors=$ $auctioneer$; $creditor=$ $auction\text{-}house$;
$content=$ $TP(setAskPrice(DutchAuction.LastPrice\text{-}\kappa),$
$[time\text{-}of(e_{start}), time\text{-}of(e_{end})], \exists, \bot)$;
$e_{start}=$ $TimeEvent(DutchAuction.timeLastPrice + \delta)$;
$e_{end}=$ $TimeEvent(time\text{-}of(e_{start}) + \lambda)$;
$conditions=$ $DutchAuction.offer.value = null$;
$a\text{-}sanctions=$ $TP(pay(ask\text{-}price, interaction\text{-}system)$;
$[time\text{-}of(e), time\text{-}of(e) + h], \exists, \bot)$;
$p\text{-}sanctions=$ $Auth(interaction\text{-}system, setAskPrice(value))$,
$Auth(interaction\text{-}system, ChangeRep(auctioneer, value))$;

where variable e refers to the event that happens if the commitment generated at run-time by this norm is violated.

5. CONCLUSIONS

An important feature of our proposal, with respect to other ones [1, 3, 6, 10, 12], is that, using the construct of a commitment, it gives a uniform solution to two crucial problems: the specification of the semantics of norms and the definition of the semantics of an Agent Communication Language. Therefore a software agent able to reason on one construct is able to reason on both communicative acts and norms. The innovative aspects of our proposal are the definition of different types of sanctions and of the operational mechanisms for monitoring the behavior of the agents and reacting to commitment violations. Thanks to their transformation into commitments, it is possible to apply certain norms (whose activation event may happen many times) more than once in the life of the system. Regarding the treatment of sanctions our model is more in-depth with respect to other proposals [10, 12, 7] because we distinguish the actions of the debtors from the actions of the other agents that are entitled to react to violations.

6. REFERENCES

[1] A. Artikis, M. Sergot, and J. Pitt. Animated Specifications of Computational Societies. In C. Castelfranchi and W. L. Johnson, editor, *AAMAS 2002*, pages 535–542. ACM Press, 2002.

[2] G. Boella and L. van der Torre. Contracts as legal institutions in organizations of autonomous agents. In V. Dignum et al., editor, *AAMAS 2004*, pages 948–955. IEEE Computer Society, 2004.

[3] M. Esteva, J. Padget, and C. Sierra. Formalizing a language for institutions and norms. In J.J.Meyer and M.Tambe, editors, *Intelligent Agents VIII: ATAL2001*, volume LNCS 2333, pages 348–366. Springer, 2002.

[4] N. Fornara, F. Viganò, and M. Colombetti. Agent communication and artificial institutions. *Autonomous Agents and Multi-Agent Systems*, 14(2):121–142, 2007.

[5] N. Fornara, F. Viganò, M. Verdicchio, and M. Colombetti. Artificial institutions: A model of institutional reality for open multiagent systems. *Artificial Intelligence and Law*, July 26, 2007.

[6] A. Garcia-Camino, P. Noriega, and J. A. Rodriguez-Aguilar. Implementing norms in electronic institutions. In *AAMAS 2005*, pages 667–673, New York, NY, USA, 2005. ACM Press.

[7] D. Grossi, H. Aldewereld, and F. Dignum. Ubi lex, ibi poena: Designing norm enforcement in e-institutions. In V. Dignum et al., editor, *COIN II*, volume LNCS 4386, pages 101–114. Springer, 2007.

[8] A. Jones and M. J. Sergot. A formal characterisation of institutionalised power. *Journal of the IGPL*, 4(3):429–445, 1996.

[9] L. Kagal and T. Finin. Modeling Conversation Policies using Permissions and Obligations. *Autonomous Agents and Multi-Agent Systems*, 14(2):187–206, 2007.

[10] F. López y López, M. Luck, and M. d'Inverno. A Normative Framework for Agent-Based Systems. In *Proc. of the 1st Int. Symp. on Normative MAS*, 2005.

[11] Y. Moses and M. Tennenholtz. Artificial social systems. *Computers and AI*, 14(6):533–562, 1995.

[12] J. Vázquez-Salceda, V. Dignum, and F. Dignum. Organizing multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 11(3):307–360, Nov 2005.

[13] P. Yolum and M. Singh. Reasoning about commitment in the event calculus: An approach for specifying and executing protocols. *Annals of Mathematics and Artificial Intelligence*, 42:227–253, 2004.