Normative Agent Reasoning in Dynamic Societies

Fabiola López y López
Facultad de Ciencias de la Computación
Benemérita Universidad Autónoma de Puebla
Puebla, Pue. México
fabiola.lopez@siu.buap.mx

Michael Luck
Electronics and Computer Science
Southampton University
Southampton, UK
mml@ecs.soton.ac.uk

Mark d'Inverno
Westminster University
London, UK
dinverm@westminster.ac.uk

Abstract

Several innovative software applications such as those required by ambient intelligence, the semantic grid, e-commerce and e-marketing, can be viewed as open societies of heterogeneous and self-interested agents in which social order is achieved through norms. For agents to participate in these kinds of societies, it is enough that they are able to represent and fulfill norms, and to recognise the authority of certain agents. However, to voluntarily *be part of a society or to* voluntarily *leave it, other characteristics of agents are needed. To find these characteristics we observe that on the one hand, autonomous agents have their own goals and, sometimes, they act on behalf of others whose goals must be satisfied. On the other, we observe that by being members, agents must comply with some norms that can be in clear conflict with their goals. Consequently, agents must evaluate the positive or negative effects of norms on their goals before making a decision concerning their social behaviour. Providing a model of autonomous agents that make this kind of norm reasoning is the aim of this paper.*

1. Introduction

Agent technology continues to contribute to the development of applications required in ambient intelligence, the semantic grid, electronic commerce and electronic marketing [6, 15, 16], all of which require open societies of heterogeneous and self-interested components. This is true if the agents are autonomous and can chose whether to be part of such societies *voluntarily*. In order to avoid the problems that might potentially arise due to conflicts between the self-interested agents, each representing the particular interests of a human agent, *norms* can provide the required means to regulate the behaviour of such participants [2, 3].

If agents are to be able to participate in a society regulated by norms, it is enough that they can represent and fulfill norms, and recognise the authority of certain agents. Moreover, in order to make decisions about whether or not they voluntarily join or leave such a society, we argue that these agents will need to have particular characteristics that are additional to autonomy. Autonomous agents, which can weigh the advantages of competing and conflicting goals, can therefore take balanced and informed decisions on joining societies whose norms may sometimes hinder the satisfaction of the agent's individual goals. This paper presents a model for such normative autonomous agents.

The autonomy needed to take decisions regarding norms leads us towards the possibility of norm infringement. This negative social behaviour is often found in humans, but may be an undesirable characteristic for computational entities. Some might even argue that one of the purposes of norms is to avoid conflicts among agents and to achieve coordination, so that conflicts of interests would arise immediately if agents were allowed to violate rules. We agree with this position but argue that although norms avoid conflicts between agents, they are the cause of some conflicts and, consequently, their infringement is an important issue to investigate. In our view, any adequate model of norms for autonomous agents must clearly address the consequences of agents willfully violating norms, for three reasons: first because individual goals can conflict with society norms; second because the norms of a society may themselves conflict in some situations; and third because the agent may be a member of more than one society.

To address how the agent should deal with the first of these problems, some authors advocate socially responsible agents who choose society goals over individual goals [5]. However, we argue that any adequate account of such sit-

uations needs to consider agents who make decisions between conflicting goals based on their *motivations*, where agents may prefer to suffer the consequences of their actions when satisfying a goal that conflicts a norm. Agents with the ability to autonomously determine which norms to fulfill, and which societies to be a part of, are clearly necessary if we wish computational entities to automatically create open societies with others. We refer to such entities as *autonomous normative agents*, and in this paper we take an existing model of agents and develop it to understand how to construct agents that can autonomously reason about norms. This approach differs from other approaches where new architectures are developed from scratch [1], or which are only concerned with norm compliance[7]. In Section 2, the basic foundational components are introduced, while Section 3 considers the types of reasoning involved to decide when to join, stay in and leave societies. Then, we develop an account of how agents can reason about why they should comply with norms, before concluding.

2. Normative Framework

We build upon our existing SMART agent framework [8] in which motivations play the key role for understanding autonomy. In the framework, an *attribute* represents a perceivable feature of the agent's environment, which can be represented as a predicate. An environmental state is then a set of attributes. Subsequently, *actions* are discrete events that can change the state of environments, *goals* represents environmental properties that an agent wishes to bring about, and *motivations* are desires or preferences that affect the outcome of the reasoning intended to satisfy an agent's goals. We assume that each agent has a unique name that allows us to differentiate one agent from another. Definitions in the Z specification language are given below.

$$[Attribute, Motivation, AgentName]$$
$$EnvState == \mathbb{P}_1\ Attribute;\ Goal == \mathbb{P}_1\ Attribute;$$
$$Action == EnvState \rightarrow EnvState$$

In SMART, *agents* are described as entities with *attributes* representing their permanent features, *capabilities* as actions that can be performed, and a set of current *goals* to bring about. *Autonomous agents* are defined as agents with *motivations* representing the preferences that drive behaviour and the adoption and creation of goals. By omitting details not relevant here, and introducing *beliefs* as the agent's internal representation of its environment, an autonomous agent can be formally represented as follows.

$$\begin{array}{|l} AutonomousAgent \\ \hline attributes : \mathbb{P}_1\ Attribute;\ name : AgentName \\ capabilities : \mathbb{P}_1\ Action \\ goals : \mathbb{P}_1\ Goal;\ beliefs : \mathbb{P}_1\ Attribute \\ motivations : \mathbb{P}_1\ Motivation \\ \hline \end{array}$$

In general, agents have different sets of motivations so that their individual preferences towards particular goals are different. In order to model this, we introduce an *importance* value of a set of goals, with respect to the current motivations, such that the greater this value the more important the goals. Explaining how this value can be obtained is beyond the scope of the paper, but we assume that there is a function for this purpose. Note that since this value depends on an agent's motivations, any change to them can lead to changes in an agent's preferences. Thus, an agent could decide to enter a society in order to satisfy an important goal and, after some time, it could then decide to leave the society if the importance of the goal decreases.

$$imp : (\mathbb{P}\ Motivation \times \mathbb{P}\ Goal) \rightarrow \mathbb{N}$$

Next, we introduce a model of *norms*, the artefacts within a society that influence the behaviour of its members. Norms can be characterised by several different aspects. First, norms must be complied with by a set of *addressee* agents in order to *benefit* another (possibly empty) set of agents. They specify what ought to be done and thus include *normative goals* that must be satisfied by addressees. Sometimes, these normative goals must be directly intended, while at other times their role is to inhibit specific goals (as in the case of prohibitions). Clearly, norms may only be applicable in certain situations, and their activation therefore depends on a *context*. However, there may also be *exceptions* when agents are not obliged to comply. Lastly, norms may suggest certain *punishments* to impose on those addressees who do not satisfy the normative goals and, possibly, a set of *rewards* when they do. Further details can be found in [12], in which we also show how different kinds of norms, ranging from obligations and prohibitions to social commitments and social codes, can be represented, but the basic model is given below.

$$\begin{array}{|l} Norm \\ \hline addressees, beneficiaries : \mathbb{P}\ AutonomousAgent \\ context, exceptions : EnvState \\ ngoals, rewards, punishments : \mathbb{P}\ Goal \\ \hline addressees \neq \varnothing \land context \neq \varnothing \\ \hline \end{array}$$

Now, in order to determine if a norm has been *fulfilled*, the satisfaction of its associated normative goals must be verified. This is true if the normative goals are a *logical consequence* of the current environmental state. We omit the formal details of this standard operator.

A *normative agent* is an agent whose behaviour is partly determined by obligations it must comply with, prohibitions that limit the kind of goals that it can pursue, social commitments that have been created during its social life and social codes that may not carry punishments, but whose fulfillment could represent social satisfaction for the agent. Moreover, autonomous agents can decide whether to adopt or ignore

norms. Here, we simply define a normative agent as an autonomous agent with adopted norms; the mechanisms for controlling behaviour are specified later.

$$\begin{array}{|l} \textit{NormAgent} \\ \hline AutonomousAgent \\ norms : \mathbb{P}\, Norm \\ \hline \end{array}$$

Then, a *normative multi-agent system (NMAS)* is defined as a set of normative agents ($members$) and the set of all norms ($systemnorms$) that govern each member. Some of the norms are legislative ($lgns$), while others punish non-compliance ($enfns$) or reward compliance ($rwns$). The authorities ($authorities$) of a NMAS are defined as all the addresses of either a legislation, an enforcement or a reward norm. Details of this can be found elsewhere [13].

$$\begin{array}{|l} \textit{NMAS} \\ \hline members : \mathbb{P}\, NormAgent \\ systemnorms, lgns, enfns, rwns : \mathbb{P}\, Norm \\ authorities : \mathbb{P}\, NormAgent \\ \hline \forall\, ag : members \bullet \\ \quad ag.norms \cap systemnorms \neq \varnothing \\ \hline \end{array}$$

3. Autonomous Membership

In accordance with our notion of autonomy, autonomous agents must express their preferences for being part of a particular relationship, group, organisation or society. Thus, agent motivations are key to understanding why agents join and stay in a society, why agents recognise the power and authority of others, and why they adopt and comply with the norms of a society. That is, agents join new societies as a means of achieving some of their individual goals or of achieving them more effectively. As long as this is the case, the rational choice would be to stay and therefore respect authority and norms. Software agents that search information in large private databases must agree, for instance, to respect confidentiality and copyright norms, before being allowed to access the required information.

Now, as members of a society, agents can also acquire certain responsibilities, which may not be dismissed as soon as they achieve their goals. For instance, an agent that joins a credit bureau to get money cannot leave the bureau until it fulfills its commitment to repay the money it borrowed. The following subsections are aimed at modelling an agent's decisions to enter and remain in a society.

3.1. Joining a Society

In general, autonomous agents join societies because some of their goals can be satisfied, or satisfied more easily. However, it is clear that some goals may be hindered by the society's norms. Moreover, an agent receives advantages and disadvantages from the direct application of norms; *contributions* may be received from the norm-based responsibilities of others, and *responsibilities* may be acquired by activated norms in which they are addressed. Clearly, in order to decide whether to be part of a society, the advantages and disadvantages must be weighed against each other.

In order to specify this assessment, we first define the set of all norms that an agent has to comply with, which we call *relevant* norms.

$$\begin{array}{|l} relevant : (NormAgent \times NMAS) \to \mathbb{P}\, Norm \\ \hline \forall\, ag : NormAgent;\ nmas : NMAS; \\ nms : \mathbb{P}\, Norm \bullet relevant\ (ag, nmas) = nms \\ \quad \Leftrightarrow (\forall\, n : nms \bullet (ag \in n.addressees \\ \quad\ \wedge\ n \in nmas.systemnorms)) \\ \end{array}$$

The next functions extract normative goals, punishments and rewards from a set of norms, respectively. Only details of the first of these functions are provided; the others are defined similarly.

$$\begin{array}{|l} normgoals, punishgoals, rewardgoals : \\ \quad \mathbb{P}\, Norm \to \mathbb{P}\, Goal \\ \hline \forall\, ns : \mathbb{P}\, Norm \bullet \\ normgoals\ ns = \bigcup \{n : ns \bullet n.ngoals\} \\ \end{array}$$

The *responsibilities* of an agent in a society are those goals that must be satisfied by the agent whilst it is a member.

$$\begin{array}{|l} responsibilities : (NormAgent \times NMAS) \\ \quad \to \mathbb{P}\, Goal \\ \hline \forall\, ag : NormAgent;\ nmas : NMAS \bullet \\ \quad responsibilities(ag, nmas) = \\ \quad\quad normgoals\ (relevant\ (ag, nmas)) \\ \end{array}$$

Agents can obtain contributions to their goals either as beneficiaries of current system norms or from the rewards of the norms they are currently fulfilling. The function $nbenefits$ determines all the norms in a society that benefit the agent, while $contributes$, determines those goals for which the agent finds some benefit from norms. The details are omitted here due to space constraints.

$$\begin{array}{|l} nbenefits : (NormAgent \times NMAS) \to \mathbb{P}\, Norm \\ contributes : (NormAgent \times NMAS) \to \mathbb{P}\, Goal \\ \end{array}$$

An agent then needs to evaluate how their *responsibilities to* and their *contributions from* a society compare. Not only must contributions provide some benefits for their *important* goals, but the new responsibilities must not hinder goals that are more important than those goals that benefit. To formalise this, we declare a function which, given two sets of goals, generates those goals in the first set that are hindered by the goals of the second. We define an analogous function for benefits.

$$\begin{array}{|l} hinder, benefit : (\mathbb{P}\, Goal \times \mathbb{P}\, Goal) \to \mathbb{P}\, Goal \\ \end{array}$$

We define a *Society Agent* as a normative agent that has a model of the societies of which it is a member as well as maintaining a model of itself.

$$
\begin{array}{|l}
\hline
SocietyAgent \\
NormAgent \\
self : NormAgent \\
societies : \mathbb{P}\, NMAS \\
\hline
self.name = name \\
\forall\, s : societies \bullet self \in s.members \\
\hline
\end{array}
$$

Now, every time an agent decides to join a new society, the evaluation of both its responsibilities and the contributions it can receive should be calculated. Agents evaluate the goals that can be *hindered* by their responsibilities and the goals that can *benefit* from the contributions they receive from the society. Then, the goals that benefit from society contributions must be more important than the goals hindered by an agent's responsibilities. We call this constraint the *social satisfaction* condition which, if fulfilled, enables an agent to enters the society, adopting the corresponding society's norms. However, this does not mean that these norms will be complied with, since the motivations of an agent might lead it to drop a norm.

The process that represents an agent's decision to enter a new society is represented in the $JoinSociety$ schema, in which the $new?$ variable represents the society that the agent is considering. The first predicate states that the agent is not currently a member of this society. The second predicate is the social satisfaction condition evaluated in the new society. The third predicate represents the agent accepting the society by including it in the set of societies to which it belongs. However, we do not update the global system state to record the fact that not only does the agent model itself as being part of this society but that it has actually joined this society. Finally, the last predicate represents the agent adopting the norms of the accepted society.

$$
\begin{array}{|l}
\hline
JoinSociety \\
\Delta SocietyAgent \\
new? : NMAS \\
\hline
new? \notin societies \\
\textbf{let}\ scgs == contributes(self, new?) \bullet \\
\textbf{let}\ args == responsibilities(self, new?) \bullet \\
\textbf{let}\ ms == self.motivations \bullet \\
\quad (imp(ms, benefit\ (goals, scgs)) \geq \\
\quad\ imp(ms, hinder\ (goals, args))) \\
societies' = societies \cup \{new?\} \\
norms' = norms \cup relevant\ (self, new?) \\
\hline
\end{array}
$$

3.2. Staying in a Society

Once agents are in a society, the satisfaction of their important goals is not the only reason they remain there. Humans, for example, do not emigrate to other societies for several reasons [2] such as not being aware of other societies, being unable to predict the benefit of joining a society, being under threat not to leave, having moral commitments to fulfill, having goals satisfied in the society, enjoying relationships with other members of a group, and so on.

In our model, we divide the reasons for remaining into two classes relating to an agent's goals and to its relationships. The first class corresponds to those reasons that cause the agent to enter the society. As long as important goals continue to be satisfied and their responsibilities do not hinder these important goals, agents will stay. The $StayForGoals$ schema below specifies this situation. It represents a normative agent that has entered a society from which norms have been adopted. This agent has a model of the society represented by $society$. The predicate states that the *social satisfaction* condition is currently satisfied.

$$
\begin{array}{|l}
\hline
StayForGoals \\
NormAgent \\
self : NormAgent \\
society : NMAS \\
\hline
self \in society.members \\
\textbf{let}\ scgs == contributes\ (self, society) \bullet \\
\textbf{let}\ args == responsibilities\ (self, society) \bullet \\
\textbf{let}\ ms == self.motivations \bullet \\
\quad (imp\ (ms, benefit\ (goals, scgs)) \geq \\
\quad\ imp\ (ms, hinder\ (goals, args))) \\
\hline
\end{array}
$$

In the second group of reasons, an agent assesses its relationships with other agents; it can decide to stay in a society in any of the following cases.

- The agent has already decided to comply with norms but their fulfillment has not yet occurred.
- The agent feels obliged to reciprocate to some agents in the society.
- The agent is part of a group of supportive agents and one of them, which is also a member of the society, needs its help.
- The agent is being coerced by a member of the society to remain there.

In the first case, the agent is simply being consistent with the normative decisions it has made. In the last three cases, the agent recognises special relationships with some other members of the society. We discuss each case separately in the remainder of this section.

An agent stays in a society when it has decided to comply with norms that have not yet been fulfilled. Here, the agent shows its respect for the commitments it has with other agents. This case is formalised in the $StayingtoComply$ schema where $intended$ represents those norms the agent has decided to comply with. The mechanism for selecting these norms is described in Section 4. The first predicate states that the agent is currently a member of the society, the second states that there are some norms of the society that the agent has decided to comply with (i.e. they are intended norms), and the third states that the agent believes that one of those norms has not yet been fulfilled.

$$
\begin{array}{|l}
\hline
\textit{StaytoComply} \\
\textit{NormAgent} \\
\textit{self} : \textit{NormAgent} \\
\textit{intended} : \mathbb{P}\, \textit{Norm} \\
\textit{society} : \textit{NMAS} \\
\hline
\textit{self} \in \textit{society.members} \\
(\textit{intended} \cup \textit{society.systemnorms}) \neq \varnothing \\
\exists\, n : \textit{intended} \bullet \\
\quad (n \in (\textit{intended} \cup \textit{society.systemnorms}) \land \\
\neg\, \textit{fulfilled}(n, \textit{beliefs})) \\
\hline
\end{array}
$$

Reciprocating actions has been considered as one of the key aspects underlying society cohesion [10]. Agents that have worked in support of the goals of others generally expect to receive reciprocal benefits, even if not explicitly agreed. This represents an ethical matter in which agents must show their gratitude to others, and could offer a way to increase *trust* between them. Since adoption of goals is made formal through commitments [11], and we are considering commitments as types of norms, an agent can determine if it must reciprocate to another agent as follows: first, there is a norm that has been complied with whose benefits were enjoyed by the first agent; second, the norm was complied with by the second agent; third, the norm did not include either punishments or rewards. This last condition is very important because it is the difference between doing something by being forced (coerced or rewarded) or by being just helpful. We specify the situation in which an agent remains in a society in order to reciprocate in this way, in $\textit{StaytoReciprocate}$. The first predicate states that the agent is currently a member of the society, the second that it believes that there is an agent in its society that complied with a norm from which it has benefited and, further, that this agent was not forced to do so.

$$
\begin{array}{|l}
\hline
\textit{StaytoReciprocate} \\
\textit{NormAgent} \\
\textit{self} : \textit{NormAgent} \\
\textit{society} : \textit{NMAS} \\
\hline
\textit{self} \in \textit{society.members} \\
\exists\, a : \textit{society.members} \bullet \\
(\exists\, n : \textit{society.systemnorms} \bullet \\
\quad (a \in n.\textit{addressees} \land \\
\quad \textit{fulfilled}(n, \textit{beliefs}) \land \\
\quad \textit{name} \in n.\textit{beneficiaries} \land \\
\quad n.\textit{rewards} = \varnothing \land n.\textit{punishments} = \varnothing)) \\
\hline
\end{array}
$$

Support relationships enable small groups to work well. Agents are empowered because they recognise the potential for unconditional help from others without any notion of reciprocation. We call this a *supportive group*. The way in which these groups are created can range from a design decision to a complex on-line mechanism in which agents voluntarily decide to group together, but that is beyond the scope of this paper. Here, we simply assume that each agent has the means to identify such a group of normative agents that must be able to comply with any commitments that benefit other agents in the group. An agent stays in a society if it is part of a supportive group and one of the members in both the same society and group needs its help. We use the $\textit{StayWithFriends}$ schema to formalise this situation, which now includes the notion of the group. The agent is a member of the group and the group is a subset of the society. Further details are omitted here but can be specified in a similar manner to the schema above.

$$
\begin{array}{|l}
\hline
\textit{StayWithFriends} \\
\textit{NormAgent} \\
\textit{group} : \mathbb{P}\, \textit{NormAgent} \\
\textit{self} : \textit{NormAgent} \\
\textit{society} : \textit{NMAS} \\
\hline
\textit{self} \in \textit{group} \\
\textit{group} \subseteq \textit{society.members} \\
\hline
\end{array}
$$

Even when the *social satisfaction* condition is not being fulfilled, an agent may stay in a society if another society member is able to hinder a goal of the first agent; this is more likely to occur if the first agent leaves. This means that although an agent's responsibilities are more than the social contributions the agent can receive, there is a more important goal that can be hindered if it decides to abandon the society. The formal representation of this is given in the $\textit{StaybyCoercion}$ schema. It describes firstly that the social satisfaction condition is not being fulfilled. Secondly, that the agent believes that it has a goal that can be hindered by another member of the society. Thirdly, that such a goal is more important than the goals hindered by the agent's responsibilities. Its final predicate represents the fact that if the agent is not a member of the society this implies that the goal will not be among its goals.

$$
\begin{array}{|l}
\hline
\textit{StayingbyCoercion} \\
\textit{NormAgent} \\
\textit{self} : \textit{NormAgent} \\
\textit{society} : \textit{NMAS} \\
\hline
\textit{self} \in \textit{society.members} \\
\textbf{let}\ \textit{scgs} == \textit{contributes}\ (\textit{self}, \textit{society}) \bullet \\
\textbf{let}\ \textit{args} == \textit{responsibilities}\ (\textit{self}, \textit{society}) \bullet \\
\textbf{let}\ \textit{bgoals} == \textit{benefit}\ (\textit{goals}, \textit{scgs}) \bullet \\
\textbf{let}\ \textit{hgoals} == \textit{hinder}\ (\textit{goals}, \textit{args}) \bullet \\
\textbf{let}\ \textit{ms} == \textit{self}.\textit{motivations} \bullet \\
(\textit{importance}(\textit{ms}, \textit{bgoals}) < \\
\quad \textit{importance}(\textit{ms}, \textit{hgoals}) \land \\
(\exists\, g : \textit{goals} \bullet (\exists\, ag : \textit{society.members} \bullet \\
\quad (\exists\, g_1 : ag.\textit{goals} \land \textit{hinders}(g_1, g)))) \land \\
\quad \textit{importance}\ (\textit{ms}, \{g\}) \geq \\
\quad \quad \textit{importance}\ (\textit{ms}, \textit{hgoals}) \land \\
\quad \textit{self} \notin \textit{society.members} \Rightarrow g \notin \textit{goals}))) \\
\hline
\end{array}
$$

All these cases can be combined through logical disjunctions to represent an agent's decisions to stay in a society

due to the relationships (or ties) it has with other agents in the society as follows.

$$StayForTies == StaybyCoercion \lor StayWithFriends \lor StaytoReciprocate \lor StaytoComply$$

3.3. Adopting new Norms

Modelling agents able to adopt new norms autonomously is an important step towards understanding dynamic societies in which changes in current legislation might occur, society members are not necessarily predetermined, and where relationships between members are created and destroyed dynamically. Enabling agents to adopt new norms allows both the independent design of these agents (because they do not need prior knowledge of the norms they must fulfill), and the possibility for agents to join or leave a society without changing their internal design. In addition, since norms represent the responsibilities of agents, and norms are different in each society, agents become able to adopt different *roles* and obligations. Moreover, the ability to adopt norms enables agents to make agreements with other agents at run-time, to either adopt or delegate their goals. Initially, the process of norm adoption received little attention from the agent research community as they were considered as built-in constraints [17], but recent approaches have incorporated the notion of generation of new norms [9].

Norm adoption can be better defined as the process through which agents recognise their responsibilities towards other agents by internalising the norms that specify these responsibilities. The importance of norm adoption as a *voluntary* process has been already pointed out by many [1, 3, 4], but their research, rather than explaining why norms are adopted, describes the cases in which norms must be rejected. These cases include situations in which: the issuer is not an authority; the norms are not within the competence of an authority; addressees are not under the authority's domain; the context in which norms are issued is not appropriate; norms are issued to satisfy an authority's personal interest; or norms are not intended to be beneficial for the group. However, not all of these can be taken as general conditions to reject norms. For instance, recognising when norms are issued to satisfy the personal interests of the issuer is not an easy task and, although this might be important for societies in which the primary objective is the equality of the members, it is too restrictive for other kinds of societies or groups. For example, suppose that a businessman wants to create a private enterprise, and one of his goals is to obtain profit. The majority of the enterprise's norms will be issued in order to guarantee the achievement of this goal. Although the norms represent the businessman's interests, employees adopt them, and as long as they want to remain in the organisation. This suggests that the motives for issuing a norm do not always coincide with the motives for adopting the norm, and a balance of interests must exist between issuers and addressees of a norm.

The concept of *authority* refers to the power assigned according to norms and accepted as legitimate by all members of a society. As long as agents want to belong to the well-defined social structure, they must adopt the norms issued by its recognised authority. Thus, for a norm to be adopted, the following conditions must be satisfied:

- the agent must recognise itself as an addressee;
- the norm must not already be adopted;
- the norm must have been issued by a recognised authority; and
- the agent must have a reason for staying in the society.

The formal representation of the first three conditions is given in the schema below. The $newnorm?$ variable and $issuer$ are given as input. The predicates state that the norm must be directed to the agent, the norm is not currently adopted, the norm is related to the issuer and the issuer is an authority of the society. If all these conditions are satisfied, then the norm is adopted by the agent.

$NormAdoption$

$$\begin{array}{l} \Delta NormAgent \\ newnorm? : Norm \\ issuer?, self : NormAgent \\ issuedby : \mathbb{P}(Norm \times NormAgent) \\ society : NMAS \\ \hline name \in newnorm?.addressees \\ newnorm? \notin norms \\ (newnorm?,\ issuer?) \in issuedby \\ issuer? \in society.authorities \\ norms' = norms \cup \{newnorm?\} \end{array}$$

The last condition represents the autonomous decision of agents. That is, to adopt a norm of its own volition, an agent must have reasons to do so. The formal representation is given below, where the schemas $StayForGoals$ and $StayForTies$ are included to represent the fact that the agent believes it has reasons to stay in the society.

$$AutonomousNormAdoption == (StayForGoals \lor StayForTies) \land NormAdoption$$

4. Autonomous Norm Compliance

To explain what might motivate an agent to dismiss or comply with a norm, and how these decisions may affects their goals, we consider the process of *autonomous norm compliance* and divide it into two separate sub-processes. In the first, the agent deliberates about whether to comply with a norm (*the norm deliberation process*). In the second, the agent updates its goals and intentions accordingly (*the norm compliance process*). Both of these processes must take into account not only the goals of agents, but also the motivations of these goals (and, therefore, their importance) and the mechanisms that the society has to avoid violation of norms such as rewards and punishments.

4.1. Norm Deliberation

To decide wether to comply with a norm, an agent must assess three things: the goals that might be hindered by satisfying the normative goals, the goals that might benefit from the associated rewards and, the damaging effects of punishments (i.e. the goals hindered due to the satisfaction of the goals associated with punishments). Since the satisfaction of some of their goals might be prevented in these cases, agents use the *importance* of their goals to make these decisions. After norm deliberation, the set of intended norms consists of those norms that are accepted to be complied with by the agent, and the set of rejected norms. More details of norm selection is provided elsewhere [14].

The state of an agent that has selected the norms it is keen to fulfill is formally represented in the $NormAgentState$ schema. This represents a normative agent with the variables representing the sets of $active$, $intended$, and $rejected$ norms at a particular point of time. There, the $conflicting$ predicate holds for a norm if and only if its normative goals conflict with any of the agent's current goals. The next three predicates state that active norms are the subset of adopted norms that the agent believes must be complied with in the current state (i.e. those norms for which the context matches the beliefs of the agent) and that, the set of active norms has already been assessed and divided into norms to intend and norms to reject. The state of an agent is consistent in that its current goals do not conflict with the intended norms and, consequently, no normative goal must be in conflict with current goals. Moreover, since rewards benefit the achievement of some goals, so that agents do not have to work on their satisfaction because someone else does, these goals must not be part of the goals of an agent. The final predicate states that punishments must be accepted and, consequently, none of the goals of an agent must hinder them.

$$
\begin{array}{|l}
\hline
NormAgentState \\
\hline
NormAgent \\
activenorms, intended, rejected : \mathbb{P}\, Norm \\
conflicting\ \_ : \mathbb{P}\, Norm \\
\hline
\forall\, n : activenorms \bullet conflicting\ n \Leftrightarrow \\
\qquad hinder(goals, n.ngoals) \neq \varnothing \\
activenorms \subseteq norms \\
\forall\, an : activenorms \bullet logcon\ (beliefs, an.context) \\
activenorms = intended \cup rejected \\
hinder(goals, normgoals\ intended) = \varnothing \\
benefit(goals, rewardgoals\ intended) \cap \\
\qquad\qquad\qquad\qquad goals = \varnothing \\
hinder(goals, punishgoals\ rejected) = \varnothing \\
\hline
\end{array}
$$

For a norm to be intended, some constraints must be fulfilled, as follows. First, the agent must be an addressee of the norm. Then, the norm must be an adopted and currently active norm, and it must not be already intended. In addition, the agent must believe that it is not in an *exception* state and, therefore, it must comply with the norm. Formally, the process to accept a single norm as input ($newnorm?$) to be complied with is specified in the $NormIntend$ schema. The first five predicates represent the constraints on the agent and the norm as described above. The sixth predicate represents the addition of the accepted norm to the set of intended norms while the set of rejected norms remains the same (final predicate).

$$
\begin{array}{|l}
\hline
NormIntend \\
\hline
newnorm? : Norm \\
\Delta NormAgentState \\
\hline
name \in newnorm?.addressees \\
newnorm? \in norms \\
newnorm? \in activenorms \\
newnorm? \notin intended \\
\neg\ logcon(beliefs, newnorm?.exceptions) \\
intended' = intended \cup \{newnorm?\} \\
rejected' = rejected \\
\hline
\end{array}
$$

To consider a norm to be rejected, the agent must be an addressee of it, the norm must be an adopted and active norm, it must not already be intended, and the agent must not be in an exception state. The process for rejecting an active norm is defined analogously to $NormIntend$ but its details are omitted here. Both processes are used in combination with different strategies in [14] to describe how agents decide whether a norm should be fulfilled.

4.2. Norm Compliance

Once agents take a decision about which norms to fulfill, *a process of norm compliance* must be started in order to update an agent's goals according to the decisions it has made. An agent's goals are affected in different ways, depending on whether the norm is intended or rejected. The cases can be listed as follows.

- All normative goals of an intended norm must be added to the set of goals because the agent has decided to comply with it.
- Some goals are hindered by the normative goals of an intended norm. These goals can no longer be achieved because the agent prefers to comply with the norm and, consequently, this set of goals must be removed from the agent's goals.
- Some goals benefit from the rewards of an intended norm. Rewards contribute to the satisfaction of these goals without the agent having to make any extra effort. As a result, those goals that benefit from rewards must no longer be considered by the agent to be satisfied, and must be removed from the set of goals.
- Rejected norms, by contrast, only affect the set of goals hindered by the associated punishments. This set of goals must be removed, and it is the way in which normative agents accept the consequences of their decisions.

To keep the model simple at this stage, we assume that punishments are always applied, and rewards are always given, though the possibility exists that agents never become either punished or rewarded. In addition, note that the set of goals hindered by normative goals can be empty if the norm being considered is a non-conflicting norm, and goals hindered by punishments or goals that benefit from rewards can be empty if a norm does not include any of them. The process to comply with the norms an agent has decided to fulfill is specified in the $NormComply$ schema. Through this process the set of goals is updated according with our discussion above.

$$
\begin{array}{|l}
\hline
NormComply \\
\Delta NormAgentState \\
\hline
\textbf{let}\ ngs == \bigcup\{gs : \mathbb{P}\ Goal \mid (\exists\, n : intended \bullet \\
\quad gs = n.ngoals)\} \bullet \\
\textbf{let}\ hngs == \bigcup\{gs : \mathbb{P}\ Goal \mid (\exists\, n : intended \bullet \\
\quad gs = hinder\ (goals, n.ngoals))\} \bullet \\
\textbf{let}\ brs == \bigcup\{gs : \mathbb{P}\ Goal \mid (\exists\, n : intended \bullet \\
\quad gs = benefit(goals, n.rewards))\} \bullet \\
\textbf{let}\ hps == \bigcup\{gs : \mathbb{P}\ Goal \mid (\exists\, n : rejected \bullet \\
\quad gs = hinder\ (goals, n.punishments))\} \bullet \\
(\ goals' = (goals \cup ngs) \setminus (hngs \cup brs \cup hps)) \\
\hline
\end{array}
$$

5. Conclusions

In this paper, we have argued that the normative behaviour of agents not only relates to the decision of whether to comply with a norm or not, but also to the decisions of whether to join, to stay or to leave a society regulated by norms. We have also argued that *societies regulated by norms* and *autonomous normative agents* are key elements of modelling open societies of heterogeneous and self-interested agents such as those required by many innovative software applications. In consequence, we have presented a model of *autonomous normative agents* that is based on a previously developed model of agents, and on previously proposed models of norms and multi-agent systems regulated by norms. To do this, we have extended the notion of *motivated autonomy* from the SMART agent framework to the normative decisions of agents. The model thus considers the reasons for agents to enter a society, to stay or leave a society depending on compliance with its norms and how the most important goals of agents are affected. We have also proposed models for the processes of autonomous norm adoption and compliance. Our model of agents, besides including many aspects not considered in existing models, also has the advantage that it can be readily incorporated into many BDI-like agent architectures. Further work will be done to address the problems of implementing the model.

References

[1] C. Castelfranchi, F. Dignum, C. Jonker, and J. Treur. Deliberative normative agents: Principles and architecture. In N. Jennings and Y. Lesperance, editors, *Intelligent Agents VI*, LNAI 1757, pages 206–220. Springer, 2000.

[2] P. Cohen. *Modern Social Theory*. Heinemann, 1968.

[3] R. Conte and C. Castelfranchi. *Cognitive and Social Action*. UCL Press, 1995.

[4] R. Conte, C. Castelfranchi, and F. Dignum. Autonomous norm-acceptance. In J. Müller, M. Singh, and A. Rao, editors, *Intelligent Agents V*, LNAI 1555, pages 319–333. Springer, 1999.

[5] M. Dastani, V. Dignum, and F. Dignum. Organizations and normative agents. In M. Shafazand and A. Tjoa, editors, *EurAsia-ICT 2002: Information and Communication Technology*, LNCS 2510, pages 982–989, 2002.

[6] D. De Roure, N. Jennings, and N. Shadbolt. The semantic grid: A future e-science infraestructure. In F. Berman, A. Hey, and G. Fox, editors, *Grid Computing: Making The Global Infrastructure a Reality*, pages 437–470. John Wiley & Sons, 2003.

[7] F. Dignum, D. Morley, E. Sonenberg, and L. Cavendon. Towards socially sophisticated BDI agents. In E. H. Durfee, editor, *Proceedings on the Fourth International Conference on Multi-Agent Systems (ICMAS-00)*, pages 111–118. IEEE Computer Society, 2000.

[8] M. d'Inverno and M. Luck. *Understanding Agent Systems*. Springer-Verlag, second edition, 2003.

[9] M. Esteva, J. Padget, and C. Sierra. Formalizing a language for institutions and norms. In J. Meyer and M. Tambe, editors, *Intelligent Agents VIII (ATAL'01)*, LNAI 2333, pages 348–366. Springer-Verlag, 2001.

[10] A. Gouldner. The norm of reciprocity: A preliminar statement. *American Sociological Review*, 25(2):161–178, 1960.

[11] N. Jennings. Commitments and conventions: The foundation of coordination in multi-agent systems. *The Knowledge Engineering Review*, 8(3):223–250, 1993.

[12] F. López y López and M. Luck. Modelling norms for autonomous agents. In E. Chávez, J. Favela, M. Mejía, and A. Oliart, editors, *Proceedings of the Fourth Mexican International Conference on Computer Science (ENC'03)*, pages 238–245. IEEE Computer Society, 2003.

[13] F. López y López and M. Luck. A model of normative multi-agent systems and dynamic relationships. In G. Lindemann, D. Moldt, and Paolucci, editors, *Regulated Agent-Based Social Systems*, pages 259–280. Springer-Verlag, 2004.

[14] F. López y López, M. Luck, and M. d'Inverno. Constraining autonomy through norms. In C. Castelfranchi and W. Johnson, editors, *Proceedings of The First International Joint Conference on Autonomous Agents and Multi Agent Systems AAMAS'02*, pages 674–681. ACM Press, 2002.

[15] M. Luck, P. McBurney, and C. Preist. *Agent Technology: Enabling Next Generation Computing (A Roadmap for Agent Based Computing)*. AgentLink, 2003.

[16] P. Maes, R. Guttman, and A. Moukas. Agents that buy and sell: Transforming commerce as we know it. *Communications of the ACM*, 42(3), 1999.

[17] Y. Shoham and M. Tennenholtz. On social laws for artificial agent societies: Off-line design. *Artificial Intelligence*, 73(1-2):231–252, 1995.