# Internal agent architecture for norm identification

Bastin Tony Roy Savarimuthu, Stephen Cranefield, Maryam A. Purvis and Martin K. Purvis

Department of Information Science, University of Otago, Dunedin, P O Box 56, Dunedin, New Zealand
(tonyr,scranefield,tehrany,mpurvis)@infoscience.otago.ac.nz

**Abstract.** Most works on norms in the multi-agent systems field have concentrated on how norms can be applied to regulate behaviour in agent societies using a top-down approach. In this work, we describe the internal architecture of an agent which identifies what the norm of a society is using a bottom-up approach. The agents infer norms without the norms being given to them explicitly. We demonstrate how the norm associated with using a park can be inferred by an agent using the proposed architecture.

## 1 Introduction

Software agents that act as proxies to real world entities need to adapt to the changing needs of environments. An example would that be of virtual worlds (e.g. SecondLife [1]). Virtual environments offer a rich and expressive environment for agent interactions. Traditionally, norms have governed the behaviour of agent interactions in a closed system. In open systems such as virtual worlds, agents instead of possessing predetermined notions of what a norm is, should be able to infer and identify norms through observing patterns of interactions and their consequences.

Recognizing the norms of a society is beneficial to an agent. This process enables the agent to know what is permissible within a society and what is not. As the agent joins and leaves different agent societies, these capabilities are essential for the agent to modify its expectations of behaviour depending upon the society it is a part of. As the environment changes, the capability of recognizing the new norm helps an agent to derive new ways of achieving its intended goals.

In this work we describe an internal agent architecture for norm identification. Using a park scenario as an example, we describe the design and implementation of the internal agent architecture which aids the agent to infer what the norms of using the park are.

## 2 Background and related work

### 2.1 Background on norms

Norms are expectations of an agent about the behaviour of other agents in the society. Norms are of interest to multi-agent system (MAS) researchers as they help in sustaining social order and increase the predictability of behaviour in the society. However, software agents tend to deviate from these norms due to their autonomy. So, the study of norms has become crucial to MAS researchers as they can build robust multi-agent systems using the concept of norms and also experiment with how norms evolve and adapt in response to environmental changes.

Due to multi-disciplinary interest in norms, several definitions for norms exist. Ullman-Margalit [2] describes a social norm as a prescribed guide for conduct or action which is generally complied with by the members of the society. She states that norms are the resultant of complex patterns of behaviour of a large number of people over a protracted period of time. Coleman [3] describes *"I will say that a norm concerning a specific action exists when the socially defined right to control the action is held not by the actor but by others"*. Elster notes the following about social norms [4]. *"For norms to be social, they must be shared by other people and partly sustained by their approval and disapproval. They are sustained by the feelings of embarrassment,anxiety, guilt and shame that a person suffers at the prospect of violating them. A person obeying a norm may also be propelled by positive emotions like anger and indignation ... social norms have a grip on the mind that is due to the strong emotions they can trigger"*.

Based on the definitions provided by various researchers, we note that the notion of a norm is generally made up of the following two aspects.

- Normative expectation of a behavioural regularity: There is a general agreement within the society that a behaviour is expected on the part of an agent (or actor) by others in a society, in a given circumstance.
- A norm spreading factor: Examples of norm spreading factors include the notion of advice from powerful leaders and the sanctioning mechanism. When an agent does not follow the norm, it could be subjected to a sanction. The sanction could include monetary or physical punishment in the real world which can trigger emotions (embarrassment, guilt etc.) or direct loss of utility resulting in the agent internalising the applicable norm to avoid future sanctions. Other kind of sanctions could include agents not being willing to interact with an agent that violated the norm or the decrease of its reputation score. Other norm spreading factors include imitation and learning on the part of an agent.

It should be noted that researchers are divided on what the differences between a norm and a convention are. Our belief is that convention is a common expectation amongst (most) others that an agent adopts a particular action or behaviour. Conventions may become norms once the non-adherence of the focal action specified by the convention is sanctioned. In this paper our concern is on *norms*.

### 2.2 Related work

Several researchers have worked on both prescriptive (top-down) and emergent (bottom-up) approaches to norms. In a top-down approach, an authoritative leader or a normative advisor prescribes what the norm of the society should be [5]. In the bottom-up approach, the agents come up with a norm through learning mechanisms [6, 7]. Researchers have used sanctioning mechanisms [8] and reputation mechanisms [9] for enforcing norms.

The work reported in this paper falls under the bottom-up approach in the study of norms. Many researchers in this approach have experimented with game-theoretical models for norm emergence [6,8]. Agents using these models learn to choose a strategy that maximizes utility . The agents do not possess the notion of a "normative expectation" in these works. Very few have investigated how an agent comes to know the norms of the society. Our objective in this work is to propose an architecture where agents can identify what the norms of the society are. Several researchers have proposed architectures for normative systems. For a comparison of these architectures refer to Neumann's article [10].

We note that our work parallels the work that is being carried out by the researchers involved in the EMIL project [11]. Researchers involved in the EMIL project [11] are working on a cognitive architecture for norm emergence. There have been some attempts to explore how the mental capacities of agents play a role in the emergence of norms.

The EMIL project aims to deliver a simulation-based theory of norm innovation, where norm innovation is defined as the two-way dynamics of an inter-agent process and an intra-agent process. The inter-agent process results in the emergence of norms where the micro interactions produce macro behaviour (norms). The intra-agent process refers to what goes inside an agent's mind so that they can recognize what the norms of the society are. This approach uses cognitive agents that examine interactions between agents and are able to recognize what the norms could be. The agents in this model need not necessarily be utility maximizing like the ones in the learning models. The agents in the model will have the ability to filter external requests that affect normative decisions and will also be able to communicate norms with other agents. Agents just employing learning algorithms lack these capabilities.

Researchers involved with the EMIL project [12, 13] have demonstrated how the norm recognition module of the EMIL-A platform works. In particular they have experimented with an imitation approach versus the norm recognition approach that they have come up with. The norm recognition module consists of two constructs, the normative board and a module for storing different types of modalities for norms (which they refer to as modals). Each modal represents a type of message that is exchanged between agents (e.g. the deontics modal refers to distinguishing situations as either acceptable or unacceptable). The normative board consists of normative beliefs and normative goals. They have shown that norm recognizers perform better than social conformers (imitating agents) due to the fact that the recognizers were able to identify a pool of potential norms while the imitators generated only one type of norm.

The work reported here differs from this work in three ways. Firstly, we have chosen "reaction" (positive and negative) to be a top level construct for identifying potential norms when the norm of a society is being shaped. We note that a sanction not only may imply a monetary punishment, it could also be an action that could invoke emotions (such as an agent yelling at another might invoke shame or embarrassment on another agent), which can help in norm spreading. Agents can recognize such actions based on their previous experience. Secondly, we identify three different sets of norms in agent's mind: suspected norms, candidate norms and identified norms. Thirdly, we demonstrate how our architecture allows for an agent to identify co-existing norms.

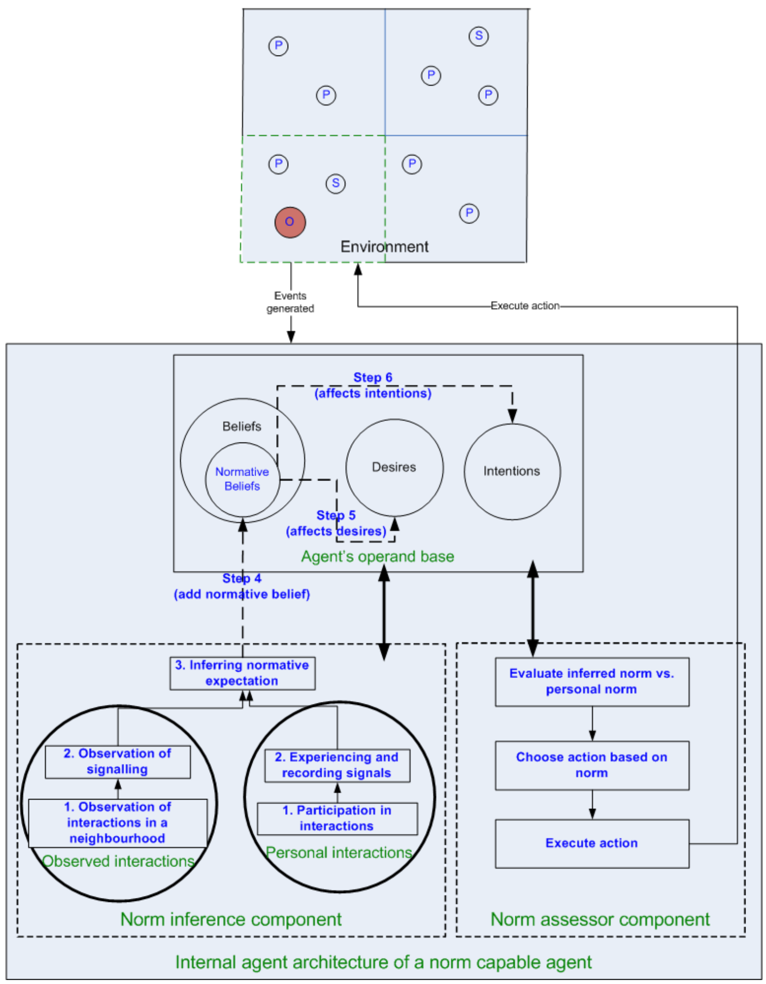


Fig. 1: Higher level architecture of norm identification

## 3 Architecture for norm identification

This section describes the normative inference architecture of an agent. The architecture provides a sequence of six steps that an agent goes through before it comes to know what a norm of the society is, as shown in Figure 1.

To understand the architecture let us assume that an agent society exists. Let us also assume that a norm does not exist to start with or only a few of the agents have a notion of what an appropriate action should be in a particular circumstance (a personal norm). In this architecture a typical agent would first observe the interactions that occur between the agents in the society. The interactions could be of two types. The first type of interaction is the one in which the agent itself is involved and is called a *personnel interaction* (an action that an agent does in an environment). The second type of interaction is an interaction between other agents that is observed by the observer agent, referred to as an *observed interaction*. The agent records these interactions. The top part of Figure 1 shows the types of agents in an agent society. An agent in the society can assume one or more of the three roles: a participant (P) that is involved in a personal interaction, an observer (O) and a signaller (S).The actions observed by an observer are of two types: regular actions and signalling actions. A regular action is an event such as an agent moving to another location in a park or sitting on a bench. Signalling actions can be thought of as special events that agents understand to be either encouraging or discouraging certain behaviour.

For example, let us assume that two agents are in a public park. One agent (A) sees another agent (B) littering the park. Agent B may choose to sanction the agent A (B nods or shakes its head in disapproval and in the worst case yells at the litterer). The observer agent (C) records the signalling that takes place between these agents. The signals can either be positive or negative and it depends on one kind of norm to another. In the case of park littering, agents might issue a negative signal when an agent litters while non-littering might be considered as a normal or routine activity for which there is no positive signalling. In our architecture, signalling is a top level entity because in normative systems it is important for an agent to have an expectation of a particular behaviour. Norms do not appear from nowhere. There might be some norm entrepreneurs or norm innovators who come up with a norm (also known as personal norm (p-norm)). Though few, these agents might sanction or reward others because they violated or followed the norm.

The third step is for the agent to infer normative expectations of a society based on noted observations and signalling. An agent correlates signalling with the observations and infers what its notion of a relevant norm in the society is. A detailed description of how the norm inference works is provided in the next section. The fourth step is to store this newly formed notion of norm in its belief set. We call the beliefs that are based on norms normative beliefs. For every signal that an agent processes, it re-evaluates its notion of the norm. Based on the inference it can modify the notion of what the norm is at any point of time which results in dynamic creation of norms. Once the agent has a norm, its desires and intentions are influenced by the norm which might affect its goals and plans (steps 5 and 6).

Once the agent has inferred what the norm it, it will then have to decide whether to follow the norm. The norm assessor component is responsible for making this decision. The agent weighs its own personal norm against the identified norm in a given circumstance and chooses an appropriate action. The emphasis of this paper is on the norm inference component.

## 4 Inferring norms in a communal park

This section describes the design and implementation of a norm identification system. The context for norms is the usage of a public park.

In many human societies there exists a norm that one should not litter a communal area such as a park. However, software agents that join open societies do not come to know of the norm of a society *a priori*. Let us assume that software agents stroll through a virtual park in environments such as SecondLife [1]. Let us imagine that the virtual park is a two dimensional grid where agents move around and enjoy the park. Agents sometimes become hungry and eat food. Some agents litter (i.e. drop the rubbish on the ground) and some agents carry the rubbish with them and drop it in a rubbish bin. The actions that can be performed by agent X are *move*, *eat* and *litter*. Some agents consider littering to be an activity that should be discouraged, so they choose to sanction other agents through actions such as yelling and shaking their heads in disapproval. We assume that an agent has a filtering mechanism which categorizes actions such as *yell* and *shake-head* as sanctioning actions. These sanctioning agents can be considered as norm entrepreneurs.

Let us assume that the agents can observe each other within a certain visibility threshold (e.g. agents can only see other agents in a 3 cell neighbourhood). Agents can either be a direct participant in interactions or observers. Some participants can be sanctioning agents . The observer records another agent's actions until it disappears from its vicinity. Whenever it encounters an action of type *sanction*, it recognizes that something has gone wrong (e.g. the action is against the personal norm of the punishing agent). When such an event occurs, the agent may become emotionally charged and perform certain sanctioning action such as yelling at the litterer or shaking its head vigorously in disapproval. Hence, an agent observing this can infer that someone involved in an interaction has violated a norm. We assume that there exists a filtering mechanism in the agent that can recognize sanctioning and rewarding actions when they occur.

Let us assume that an agent perceives other agents' actions. An event that is perceived consists of an event id, an observed action, and the agent(s) participating in that event. For example an agent observing another agent eating will have the representation of *do(1,eat,A)*. This implies that the observer believes that the first event was generated by agent $A$ which performs an action *eat*. A sample representation of events observed by an agent is given below.

$$\begin{pmatrix} do(1, eat, A) \\ do(2, litter, A) \\ do(3, move, B) \\ do(4, move, A) \\ do(5, sanction, B, A) \end{pmatrix}$$

Event 5 is a sanctioning event where agent B sanctions agent A. An agent records these events in its belief base. The agent has a filtering mechanism, which identifies signalling events. We can consider the filtering mechanism to be a black box that recognizes an emotionally charged event such as yelling and shaking head in disapproval and categorizes those actions to be sanctions [1]. When a sanctioning event occurs, it triggers the invocation of the norm inference module of the agent. It should be noted that signalling events can both be positive (e.g. rewards) and negative (e.g. sanctions). In this work, we have focused on the latter type of signalling.

Figure 2 shows the architecture of the norm inference component of an agent. The following sub-sections describe the four sub-components of the norm inference component.

### 4.1 Creating event-episodes

Agents record other agents actions in their memory. Let us assume that there are three agents A,B and C. Agent A eats, litters and moves while agent B moves and then sanctions. Agent C observes these events and categorizes them based on which agent was responsible for creating an event. $\{A\}$ followed by right arrow ($\rightarrow$) indicates the categorization of events performed by agent $A$ as observed by agent $C$. A hyphen separates one event from the next.

$$\begin{pmatrix} \{A\} \rightarrow do(1, eat, A) - do(2, litter, A) - do(4, move, A) \\ \{B\} \rightarrow do(3, move, B) - do(5, sanction, B, A) \end{pmatrix}$$

When a sanction occurs, an observer agent extracts the sequence of actions from the recorded history that were exchanged between the sanctioning agent and the sanctioned agent. In the example shown above, the observer infers that something that agent A did may have caused the sanction. It could also be that something agent A failed to do might have caused a sanction. In this work we concentrate on the former of the two. Agent C then extracts the following sequence of events that take place between A and B based on the information retrieved from its history.

$$\{A, B\} \rightarrow eat(1, A) - litter(2, A) - move(4, A) - sanction(5, B, A)$$

[1] Recognizing and categorizing a sanctioning event is a difficult problem. In this paper we assume such a mechanism exists (e.g. based on an agent's past experience)

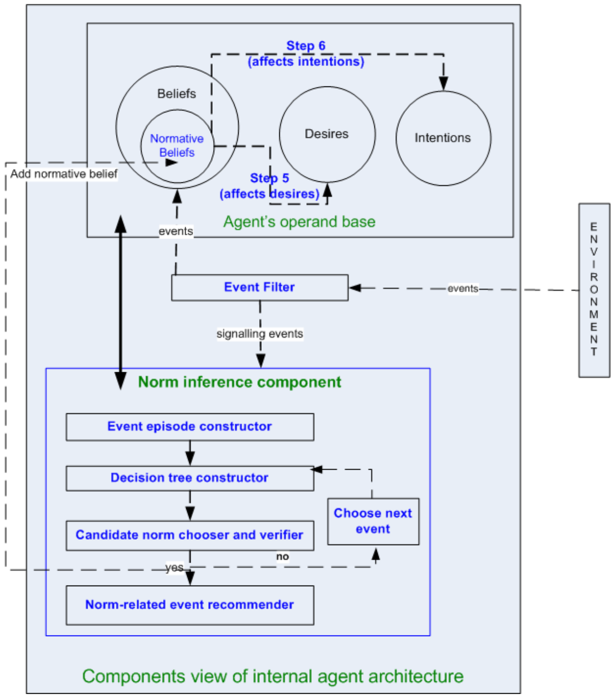


Fig. 2: Architecture of the norm inference component

To simplify the notation here afterwards only the first letter of each event will be mentioned (e.g. $e$ for eat). The event episode for interactions between agents A and B shown above will be represented as

$$\big(\{A, B\} \rightarrow e - l - m - s\big)$$

There might be a few sanctioning events at any given point of time that an agent observes. A sample list containing ten event episodes that are observed by an agent in a certain interval of time is given below.

$$\begin{pmatrix} e-l-m-s, l-e-l-s, m-e-l-s, e-l-e-s, e-l-e-s \\ l-e-l-s, e-e-l-s, m-e-l-s, e-l-m-s, e-l-e-s \end{pmatrix}$$

### 4.2 Constructing an event-tree based on conditional probability

Once the event episodes are constructed, the agent creates a tree of events that occur in all episodes based on the estimation of conditional probabilities for events that might have led to sanctioning. The mechanism for constructing a decision tree is explained below.

For calculating the conditional probabilities for events that precede a sanction, an agent follows the following steps.

1. Categorizes episodes into events belonging to different levels.
2. Constructs a conditional probability tree of sub-episodes
3. Ranks sub-episodes and chooses candidate norms for verification

**Categorizing episodes into event levels -** Based on a certain fixed number of events that precede a sanction, an agent categorizes events of an episode into certain levels (e.g. single-level events, two-level events and three-level events). Let us assume that an agent is interested in $n$ events in a sequence that precede a sanction. As an example let us consider *e-l-m-s*, which is the first episode from the sample list of ten episodes. The sequence of events that precede a sanction is *e-l-m* and hence the value of $n$ is three. A single level event (level 1) is an event that precedes a sanction (i.e. $m$). Two-level events (level 2) are the events that are a combination of two events that precede a sanction (i.e. *e-l* and *l-m*). Three-level events (level 3) are the events that are a combination of three events that precede the sanction (i.e. *e-l-m*). Let us call each entry in these levels a sub-episode.

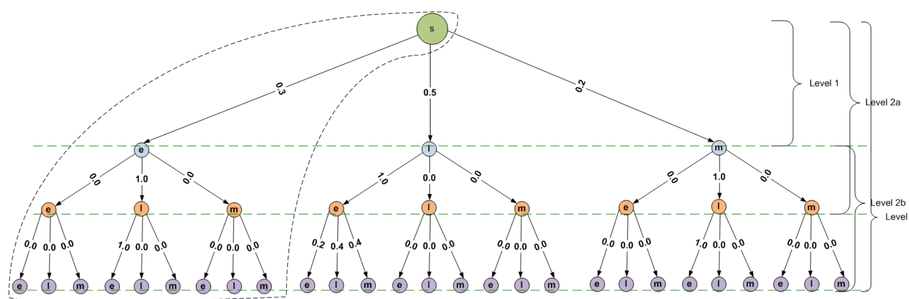


Fig. 3: Events-tree of all episodes based on conditional probability

**Constructing a tree based on conditional probability -** For each sub-episode in each level, the agent calculates the conditional probability. Sub-episodes for an episode *e-l-m* are $e$ at level 1, *e-l* and *l-m* at level 2 and *e-l-m* at level three. The conditional probability tree of the sample list of ten events as shown in Section 4.1 is given in Figure 3.

For the sake of simplicity, let us only consider those sub-episodes that end with $e$ in the region encompassed by a dashed line in Figure 3. In the sample list that consists of ten episodes, there are three episodes that end with event $e$. So, the conditional probability of event $e$ given that a sanction has occurred is p(e|s)=0.3. One of the three events (e or l or m) could have occurred before $e$. The conditional probability of $e$ occurring given that an *e-s* has occurred is p(e-e-s|e-s)=0.0 and the other two conditional probabilities are p(l-e-s|e-s)=1.0 and p(m-e-s|e-s)=0.0. Based on these, we know that p(l-e-s|s)=0.3 and the p(e-e-s|s)=0 and p(m-e-s|s)=0. Now again, three events (e or l or m) could have preceded *l*. The conditional probabilities p(e-l-e-s|l-e-s)=1.0 and p(l-l-e-s|l-e-s)=0 and p(m-l-e-s|l-e-s)=0. From these, we can infer that p(e-l-e-s|s)=0.3, p(l-l-e-s|s)=0 and p(m-l-e-s|s)=0.

At level 2, we are also interested to find out the occurrences of all episodes that are made up of two level events (indicated in figure 3 as levels 2a and 2b). Based on permutations with repetitions we know that for choosing two out of three events, there are 9 possible combinations (ee,el,em,le,ll,lm,me,ml,mm). The respective probabilities of each of these sub-episodes is 0.1,1,0,0.5,0,0.2,0.2 and 0,0.

The list given below shows the conditional probabilities of all sub-episodes that have a non-zero probability for all the three levels. We call these sub-episodes *suspected norms*. Note that for simplicity we assume that the representation of p(x|s) is p(x). Additionally, the hyphens will be omitted from the sub-episodes (e.g. *e-l-m* will be represented as *elm*).

1. p(e)= 0.3, p(l)=0.5, p(m)=0.3
2. p(ee)=0.1, p(el)=1, p(le)=0.5, p(lm)=0.2, p(me)=0.2
3. p(ele) = 0.3, p(eel)=0.1, p(lel)=0.2, p(mel)=0.2, p(elm)=0.2

**Ranking sub-episodes and selecting candidate norms -** The agent ranks sub-episodes based on these probabilities and creates a ranked list using the norm selection parameter (*ns*). An agent chooses only those sub-episodes that have conditional probabilities greater than *ns*. Elements in this subset of norms are referred to as *candidate norms*. For example, if *ns* is set to 50, the candidate norms chosen from the set of suspected norms will be *el* (100%), *l* (50%) and *le* (50%).Having compiled a set containing candidate norms, the agent passes this information to the norm verification and identification component.

### 4.3 Norm verification and identification

In order to find whether a candidate norm is a norm of the society, the agent asks another agent in its proximity. This happens in certain intervals of time (e.g. once in every 10 iterations). When two agents A and B interact, A chooses its first candidate norm and asks B whether its current norm is A's candidate norm. If true, A stores this norm in its set of *identified norms*. Otherwise, it chooses a sub-episode of the norm and enquires whether that is the norm. It is possible that B might identify the sub-episode as the norm. If not, A moves on to the second candidate norm in its list[2].

In the case of the running example, the sub-episode *el* has the highest probability for selection and it is chosen to be communicated to the other agent. It asks another agent (e.g. an agent who is the closest) whether it thinks that the given candidate norm is a norm of the society. If it responds positively, the agent

[2] Other alternative mechanisms are also possible. For example, an agent could ask for all the candidate norms from another agent and can compare them locally.

infers *prohibit*($el$) to be a norm. If the response is negative, this norm is stored in the bottom of the candidate norm list. It then asks whether the sub-episodes of *el*, which are $e$ or $l$ are the reasons for sanction. If yes, the appropriate action is considered to be prohibited. Otherwise, the next event in the candidate norm list is chosen. This process continues until a norm is found or no norm is found in which case, the process is re-iterated once a new signal indicating a sanction is generated. When one of the candidate norms has been identified as a norm of the society, the agent still iterates through the candidate norm list to find any co-existing norms.

It should be noted that an agent will have three sets of norms: suspected norms, candidate norms and identified norms. Figure 4 shows these three sets of norms. Once an agent identifies the norms of the system and finds that the norms identified have been stable for a certain period of time, it can forgo using the norm inference component for a certain amount of time. It invokes the norm inference component in regular intervals of time to check if the norms of the society have changed, in which case it replaces the norms in the identified list with the new ones[3].

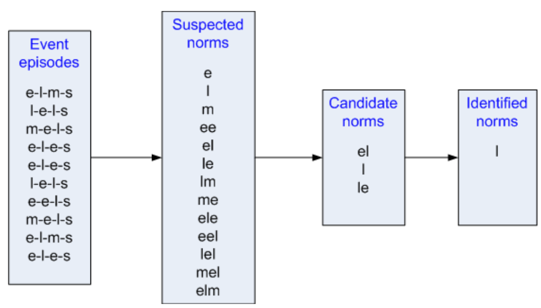


Fig. 4: Three sets of norms

### 4.4 Related event recommender

Even if the event immediately preceding a sanction was responsible for causing the sanction (e.g. event $l$), the agent would still be watchful of the event sequences that precede the sanctioned action 100% of the time (e.g. event $e$) for two reasons. One reason is that when it produces events $e$ and then $l$, it could be sanctioned. Also when other agents produce events *e-l*, then if the observer were a sanctioning agent, it may have to sanction the litterer. The purpose of the related event recommender is to recommend event episodes that occur 100% of the time preceding a sanctioning action so that the agent can be warned about impending sanctions.

[3] Alternatively, an agent can wait for certain number of sanctions to occur before it invokes the norm inference component

## 5 Experiments on norm identification

In an agent society, one or more norms can co-exist. In this section we demonstrate that the agents using our architecture are able to infer the norms of the society.

### 5.1 Scenario 1 : A society with one type of norm

We have experimented with an agent society comprising 100 agents. There are agents with three different personality types. They are learning litterers (*ll*), non-litterers (*nl*) and non-littering punishers (*nlp*). The learning litterers are litterers who learn to change their behaviour based on normative expectations inferred through the observation of interactions between agents. Non-litterers do not litter the park and non-littering punishers are the non-litterers who sanction littering because that action is against their personal norm.

There are 50 *ll* and 50 *nl* agents. Out of these 50 *nl* agents, 5 are *nlp* agents. In each iteration, an agent performs one of m,e,l or s. The agents are initialized with a uniform probability for choosing actions (p(m)=0.75, p(e)=0.25, p(l) having eaten in the previous interaction =0.5). The *nlp* agents punish other agents if they observe a littering action of an agent in the current iteration or the previous iteration with 6% probability (in both the cases). An agent stores the actions performed by other agents in its vicinity (in the current set up, a fully-connected network topology is assumed where an agent can see all other agents). We ran this experiment for 100 iterations. In each iteration an agent can perform one of the actions (*e,l,m,s*). At the end of the run every agent looks at the event history it had recorded and observes what kinds of suspected and candidate norms has emerged.

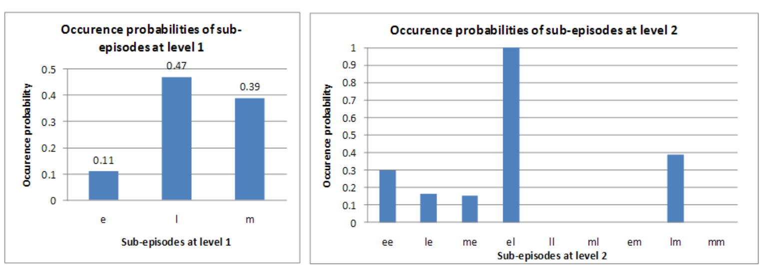


Fig. 5: Sub-episode occurrence probabilities (level 1 and 2)

It should be noted that for an episode that is made up of 3 different events, allowing permutation with repetition, 39 sub-episodes can be created (3 in level 1, 9 in level 2 and 27 in level 3). It can be observed from figures 5 and 6 that out of 39 possible sub-episodes, only a subset of sub-episodes (13 of them) happen to appear (i.e. the suspected norms). Assuming that an agent's norm selection

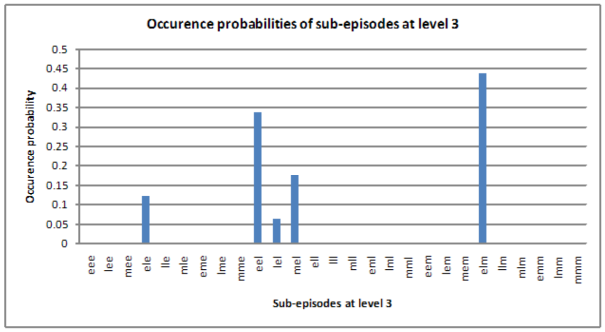


Fig. 6: Sub-episode occurrence probabilities (level 3)

threshold is 0.45 to construct the list of candidate norms, there are two such norms, which are norms against *el* and *l*. The agent then moves on to the norm verification stage which identifies the norm against littering.

### 5.2 Scenario 2 : Identification of co-existing norms in an agent society

Let us assume that there are two types of sanctioning agents, one that sanctions when an agent litters the park and the other sanctions if it sees anyone eating in the park. In these cases, our mechanism will be able to generate different sets of suspected norms. Retaining the experimental set-up used in the previous scenario, we have set the probability of a *nlp* punishing eating action to be 3% and the probability of punishing littering action to be 3%). The norm selection threshold has been set to 0.25. Occurrence probabilities of sub-episodes (i.e suspected norms) at level 2 is given in figure 7. It can be observed that there are more occurrences of events involving *e* that appear in these sub-episodes than event *l*. This is due to the set up of the system since p(e)=0.25 while p(l)=0.125. The important thing to note in this experiment is that our architecture allows for the identification of co-existing norms.

### 5.3 Scenario 3 : Identification of norms across different societies

Let us assume that there are three sections of a park. At any point of time, an agent might be present in one of these sections. Let us also assume that there are two types of sanctioning agents. One type of agents punish litterers while the other type of agents punish those who eat in the park. Assume that these types of agents are randomly placed in the three sections of a park. Our objective was to see what type of norms might emerge in these three sub-groups.

Figure 8 shows the candidate norms of three different agents that belong to three different sub-groups. The norm selection threshold was set to 0.3. It can be observed that different types of candidate norms are generated in the minds of these agents based on what they had observed in their respective agent

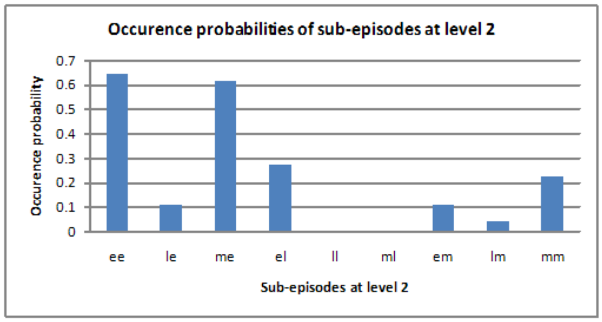


Fig. 7: Sub-episode occurrence probabilities (level 2) when two types of sanctioning agents were present

society. It can be observed that the agent from sub-group one had identified the norm against littering and the one from the third group had identified the norm against eating while the agent from the second group had identified both these norms.

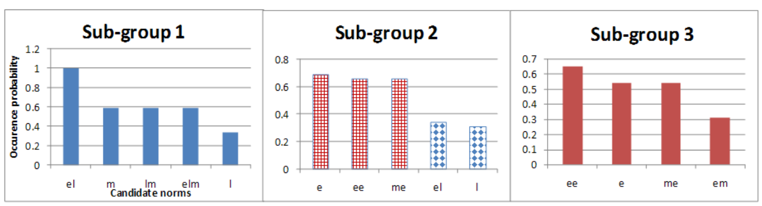


Fig. 8: Candidate norms of three agents that belong to three different sub-groups

An extension to this experiment is to allow an agent to move around in these three sections of the park and see how it accommodates the changes to its norms. Another extension will be to allow the sanctioning agents to move around which will enable dynamic change of norms in the society.

## 6 Discussion

We note that the experimental set up is simple. We have assumed that an agent considers three events ($n$=3) that precede a signal (a sanction or a reward). The value of $n$ can change and an agent being a computation machine should be able to handle a large number of possible events. Most researchers agree that there will be some form of sanction or reward once a norm is established (e.g. [3, 4]). Hence, the notion of a reaction (positive or negative action) has been considered to be a top level entity in our work. We have assumed that even when a norm is being created, the notion of sanction is important for norm identification.

Our experimental set-up can be improved in many ways. Firstly, we do not assume that there is a cost associated with sanctions. The cost for sanctions can be included in the model. Secondly, our model identifies co-existing norms. If the cost of sanctions is considered there could also be competing or conflicting norms. For example, some agents might punish other agents when they litter while some others may punish one when the littering agent is within 20 meters from the rubbish bin. When there are competing norms the society might be divided into groups. This type of dynamics will be interesting to study. Thirdly, the experiments have only made use of observational information ignoring the personal experience. We believe that the inclusion of personal experience will speed up the rate at which norms are identified. Fourthly, we have assumed that when an agent identifies a norm, it will follow the norm. Agents owing to their autonomy do not always follow the norm. An agent might have its own personal agenda and it can be an opportunistic norm follower. Autonomy of an agent needs to be addressed in the future. In our architecture, this has been encapsulated as a part of the norm assessment component which will be elaborated in a future work. Fifthly, it might not always be possible to associate sanctions or rewards with the events that immediately precede them. For example, speeding might result in a fine that is sent to an agent after a couple of days. An observer might not be able to recognize this sanction. In this work, we have only considered those norms where the sanctions can be recognized by an observer and the events that caused the sanction occurred within an immediate window of time before the sanction. Sixthly, the problem of false negatives and positives for norm identification needs to be dealt with in the future. Lastly, our work can take advantage of the work done in the data-mining field on the identification of frequent event sequences [14].

However, we believe that our work reports some advancements. Firstly, the question of "how an agent comes to find out what the norm of society is" is being dealt with by at least one other research group [11]. We have made some progress in that regard by proposing an internal agent architecture and demonstrating how an agent will identify the norms of a society. Secondly, other prominent works identify one norm that exists in the society [8, 15]. In our architecture an agent is able to identify several norms that might exist in the society. Thirdly, most works have not addressed how an agent might be able to identify whether a norm is changing in a society and how it might react to this situation. In our model, the agents will be able to identify the norm change and dynamically add, remove and modify norms. Fourthly, our architecture can be used to study norm emergence. We believe through norm identification at the agent level, we are also in the realm of addressing how norms emerge using a bottom-up approach.

## 7 Conclusions

In this paper we have explained the internal agent architecture for norm identification. Through simulations we have shown how an agent infers norms in an agent society. We have also discussed the related work and have identified issues that should be addressed in the future.

## 8 Acknowledgments

We thank the three anonymous reviewers for their insightful comments.

## References

1. Rymaszewski, M., Au, W.J., Wallace, M., Winters, C., Ondrejka, C., Batstone-Cunningham, B., Rosedale, P.: Second Life: The Official Guide. SYBEX Inc., Alameda, CA, USA (2006)
2. Ullmann-Margalit, E.: The Emergence of Norms. Clarendon Press (1977)
3. Coleman, J.: Foundations of Social Theory. Belknap Press (August 1990)
4. Elster, J.: Social norms and economic theory. The Journal of Economic Perspectives **3**(4) (1989) 99–117
5. Verhagen, H.: Simulation of the Learning of Norms. Social Science Computer Review **19**(3) (2001) 296–306
6. Shoham, Y., Tennenholtz, M.: Emergent conventions in multi-agent systems: Initial experimental results and observations (preliminary report). In: KR. (1992) 225–231
7. Sen, S., Airiau, S.: Emergence of norms through social learning. In: Proceedings of Twentieth International Joint Conference on Artificial Intelligence (IJCAI'07), AAAI Press (2007) 1507–1512
8. Axelrod, R.: An evolutionary approach to norms. The American Political Science Review **80**(4) (1986) 1095–1111
9. Castelfranchi, C., Conte, R., Paolucci, M.: Normative reputation and the costs of compliance. Journal of Artificial Societies and Social Simulation **vol. 1, no. 3** (1998)
10. Neumann, M.: A classification of normative architectures. In: Proceedings of World Congress on Social Simulation. (2008)
11. Andrighetto, G., Conte, R., Turrini, P., Paolucci, M.: Emergence in the loop: Simulating the two way dynamics of norm innovation. In Boella, G., van der Torre, L., Verhagen, H., eds.: Normative Multi-agent Systems. Number 07122 in Dagstuhl Seminar Proceedings, Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany (2007)
12. Andrighetto, G., Campenni, M., Cecconi, F., Conte, R.: How agents find out norms: A simulation based model of norm innovation. In: 3rd International Workshop on Normative Multiagent Systems (NorMAS 2008), 15-16 July, 2008, Luxembourg. (2008)
13. Campenni, M., Andrighetto, G., Cecconi, F., Conte, R.: Normal = normative? the role of intelligent agents in norm innovation. In: The Fifth Conference of the European Social Simulation Association (ESSA), University of Brescia, September 1-5, 2008. (2008)
14. Mannila, H., Toivonen, H., Inkeri Verkamo, A.: Discovery of frequent episodes in event sequences. Data Mining and Knowledge Discovery **1**(3) (1997) 259–289
15. Epstein, J.M.: Learning to be thoughtless: Social norms and individual computation. Comput. Econ. **18**(1) (2001) 9–24