****ANIMATION OF OPEN AGENT SOCIETIES****

Alexander Artikis ; Lloyd Kamara ; Frank Guerin ; Jeremy Pitt
**** Imperial College of Science, Technology and Medicine ****
**** Electrical & Electronic Engineering ****
**** Intelligent & Interactive Systems Group ****
**** {a.artikis;l.kamara;f.guerin;j.pitt}@ic.ac.uk ****

Abstract

E-commerce, Virtual Enterprises and digital media rights management are examples of important application domains where independently developed information systems must form coherent and stable "societies". Societies are considered stable if the behaviour and the interactions of the members are norm-governed. To investigate such normative behaviour we propose a new formal model of open artificial societies that focuses on externally observable states of affairs, and represents phenomena such as social constraints, social norms (obligations and permissions), communication language, social structure, ownership and society dynamics. Given the formal model we can examine whether or not societies remain stable over time. In order to further examine the evolution of societies we propose a specification for the integration of the society model with a multi-agent Animator that explicitly addresses the social issues in the simulations of artificial societies. We illustrate the functionality of the proposed Animator in a trading scenario. Our expectations are that the automation of the animation of artificial societies will enable the scaling-up of the analysis of the simulated societies to real open agent societies.

1. Introduction

E-trading, Virtual Enterprises and digital media rights management are examples of important application domains where independently developed information systems must form coherent and stable "societies". The stability of such societies is primarily based on the behaviour and interactions of the agents that form the society. Therefore, forming and designing an open agent society imposes a set of requirements. Firstly, there is a need to make the organisational and legal elements of MAS externally visible and provide formalisations of agent interactions to protect members from the actions of other members. Secondly, open societies should be neutral with respect to the internal architecture of their members. Thirdly, in a society, communication and conformance of behaviour are at least as important as intelligence.

We present a theoretical framework that addresses these requirements. Firstly, we view agent societies as normative systems and we describe and formalise agent behavior and society rules/constraints, in terms of normative relations given their role (standing) in the society. Secondly, we model agent societies based only on externally observable events, but although we mainly reason about the global state of a system, we can also make inferences about agents' mental states. Thirdly, we explicitly represent the aspects of communication, norms and agent ownership as parameters of the agent society.

We propose a tool for automation of animations of agent societies, by specifying an integration of the theoretical framework with a multi-agent Animator. The proposed Animator compiles the society model and its formally defined components, as well as the physical environment of the simulated societies for the benefit of external observers and simulated agents.

The argument followed in the paper is illustrated below. We begin by describing previous work on performing trust experiments in simulations of trading communities. Based on our experience on these experiments we stress the need for a formal framework of open agent societies. We then review background research and present our formal framework of open societies. Communication is an important component of that framework; therefore, we give a detailed description of the way we give semantics to the acl messages. Further, we present a DAI Animator and propose a specification of an integration of this Animator with the theoretical framework. Next, we briefly describe an e-commerce scenario and animate a society formed to execute a trading protocol in that scenario. Finally, we conclude that the integration of the Animator with the theoretical model is feasible, and, based on the interesting properties that were revealed in the animation of the trading protocol, we expect that by producing an automation of animation of artificial societies we will facilitate the process of deploying real open agent societies.

2. Motivation

In previous work (Witkowski et al. 2000) we investigated aspects of trust in a virtual trading community that was based on a model of an Intelligent Network (IN). This trading community was simulated in the MARINER project[1]. The design of this trading community was agent-oriented rather than system-

[1] http://www.teltec.dcu.ie/mariner

oriented in the sense that the focus was on the internal mental states of the agents and especially on the trust function of the customer and supplier agents. As a result, agents from other societies would not enter this 'closed' society due to the lack of well-defined societal constraints and legal assurances. We follow (Dellarocas et al. 2000) in that "the most successful marketplaces will be the ones that provide the best 'quality of service' guarantees in terms of fairness and assurances, while meeting such challenges such as agent heterogeneity, limited trust and potential systemic dysfunctions". Therefore, we present a theoretical formal framework for specifying and designing open agent societies from an external perspective[2] and propose a specification of a practical implementation of that framework in order to animate the execution of such societies. The theoretical framework identifies the key elements of open artificial societies and attempts to formalise them. One of these elements is the communication language which is given a denotational semantics. We review background research and then we present the theoretical framework as well as the specification of its implementation in the following sections.

2.1 Background

We review the action modality that Jones and Sergot (1993; 1996) use, the 'counts as' connective, and their definition of institutional constraints. The action modality E_a is used both for expressing that agent a brings it about/sees to it that some state of affairs holds, and for expressing that agent a performs a designated act. This logic of action ignores both the means by which a state of affairs is brought about and temporal aspects.

A special kind of a conditional relation is introduced, called "counts as" and it is used to represent the notion of institutionalised power. Within a given institution or norm-governed system, a particular action of a specified agent counts, in that institution, as a way of establishing an institutionalised fact. For example, in an auction house, the auctioneer's performing the speech act "the item x is sold" counts, in the auction house, as a way of establishing that item x is sold and, therefore, no one can bid for it anymore. The "counts as" connective is represented by $\Rightarrow_s$ where s is the given norm-governed system. The previous example can be represented as:

$$E_{auctioneer} sold \Rightarrow_{auction\_house} E_{auction\_house} Sold$$

If the auctioneer performs the *sold* speech act, then this counts, in the auction house, as the fact that this auction house sees to it that *Sold*, i.e. the item is sold.

[2] (Venkatraman and Singh 1999) argue that because of the autonomy and heterogeneity requirements of open systems such as the Internet, compliance testing cannot be based on the internal architecture of the agents.

Jones and Sergot (1996) represent the constraints on an institution (which may be, among other things, conditionals which describe relations of logical, causal, deontic and "counts as" consequence) as a necessity statement:

$$D_s(A \rightarrow B)$$

This expression can be read as "it is a constraint of the institution or norm-governed system s that if A then B". Regarding the "counts as" connective, the following schema has been adopted:

$$(A \Rightarrow_s B) \rightarrow D_s(A \rightarrow B)$$

Due to space limitations we do not present the semantic meaning of the previously described concepts; we refer the reader to (Jones and Sergot 1993; 1996) for a detailed presentation of these semantics.

3. Society Model

In this section we present a brief description of our preliminary attempt to formally model agent societies[3] and discuss ways towards a grounded semantics of this model.

3.1 Components of the Society Model

We identify some concepts that constitute an agent society or a norm-governed system. These are a set of agents, a set of rules/constraints on the society, a communication language, a set of social roles that members can play, a social state and the owners of the members. We claim that these are fundamental elements of a norm-governed system and that an agent society model is incomplete if it does not include all of these elements. What follows below is a description of these elements.

Society Constraints. We view agent societies as instances of norm-governed systems. Therefore, we need a way of representing rules and normative relations inside a society. We apply the notion of institutional constraints to agent societies in general and we use that notion in order to formalise and give semantics to the society rules. We apply the institutional constraints on societies of agents and not just on institutions. Therefore, from now on we will use the term society/social constraints or just constraints. The set of these constraints is represented in our model of agent societies or norm-governed systems by the $\Delta_{soc,t}$ set. Society constraints along with the use of deontic operators and the action modality enable us to represent, among other things, the normative relationships of agents. These constraints also describe the agent communication (ACL specifications are translated into constraints), the agent behaviour that results from the social roles that agents occupy, and the agent behaviour in general (regardless of what roles they occupy).

[3] A more detailed description of the society model can be found in (Artikis et al. 2001) and (Artikis and Pitt 2001).

ACL. One of the most important aspects of an agent society is communication. We include in our model the specifications of the semantics of the ACL that is used in the society under investigation. These specifications define the majority of the social constraints. We view the ACL semantics from an external perspective. A detailed description of these semantics is given in the next section.

Definition (social role). A *social role r* is defined as the set of preconditions (P_r) that an agent must satisfy in order to occupy that role, and as the set of constraints (Δ_r) of the form $D_s(A \rightarrow B)$ where the antecedent A or consequent B describe norms that are associated with that role (i.e. obligations, permissions), states of affairs that can be brought about by agents occupying that role, or, other states of affairs that identify the behaviour of the agents occupying that role. So, at time t a role r_t is mapped (with the use of a function l) to its preconditions and constraints, i.e. $l(r_t, t) = (P_{r,t}, \Delta_{r,t})$.

In general, given our definition of social roles, in a norm-governed system the following constraint holds:

$$D_s(preconditions(role,\ agent) \leftrightarrow PE_{agent}\ role\_of(role,\ agent))$$

An agent is permitted[4] to see to it that he occupies a role if and only if he satisfies the preconditions of that role. In addition, in a norm-governed system, the following constraint *may* be desired:

$$D_s(role\_of(role,\ agent) \rightarrow preconditions(role,\ agent))$$

I.e. it is a constraint of the system that an agent satisfies the preconditions of a role as long as he occupies that role.

Definition (state of the society). A state S_t of the society/norm-governed-system is defined as the set of propositions/states of affairs that are true at time t.

We have defined a *statechange* function which, given the current social state (S_t) and the set of the current externally observable events ($Events_t$), produces a set of propositions that hold in the next social state (S_{t+1}).

$$statechange_t : (\wp S_t \times \wp Events_t) \rightarrow \wp S_{t+1}$$

A brief description of the proposed implementation of the *statechange* function is the following: the set of propositions of the current state and the set of current events 'activate' a subset of the society constraints (we call this subset the set of active society constraints, i.e. the set of active constraints of a time point t is the set of society constraints of the form $D_s(A \rightarrow B)$ where the antecedent A is true at time t). The complete next social state (S_{t+1}) is the union of the current social state (S_t) with the set of the consequents of the active social constraints. Finally, if the propositions of the next social state 'activate' any more social constraints then the consequents of these active constraints are unified with the next social state. The latter step is followed until the next social state contains propositions that do not 'activate' any social constraints.

Ownership. Each agent in a society represents/is owned by either another individual agent, human or artificial, or an institution[5]. This notion of ownership is very important in determining the legal obligations that are associated with 'real world' transactions, i.e. the liability and responsibility for agents' actions. Agent interactions and their implications should be associated with legal obligations between the owners of the interacting agents.

Sub-societies. It is often the case that agents form sub-societies inside a greater society of agents (i.e. for the execution of a protocol). We follow the OO paradigm to represent sub-societies. Therefore, we can have a hierarchy of societies of an arbitrary depth, one nested in an other. Each sub-society inherits and possibly overrides properties like social roles, social constraints and ACL specifications from its parent society. In particular, for representation purposes, each sub-society is treated as if it was an individual society. The model of each society includes pointers to all its sub-societies. As far as social constraints are concerned, inheritance is defined in our model as the union of the set of constraints of the parent society with the set of additional constraints of the sub-society (the latter set can be empty), having resolved any inconsistencies that may exist in the union of these constraints. Inconsistencies may arise as a result of the union of the sets of constraints (we assume that each set is consistent). One way of resolving inconsistencies is the following: we remove constraints of the parent society that create inconsistencies in the union of constraints.

Definition (agent society model). An agent society *s* is modeled as:

$\Sigma_{soc,t} = (Ag_t, \Delta_{soc,t}, ACL, R_t, S_t, SubSoc_t, f, g, h, k, l)$

- $Ag_t = \{a_1, a_2, ..., a_n\}$ is a non-empty set of member agents.
- $\Delta_{soc,t}$ is a non-empty set of society constraints. $\Delta_{soc,t}$ is the union of the ACL constraints, role constraints ($acl\_constraints_t$, $role\_constraints_t \subseteq \Delta_{soc,t}$) and other constraints that are not associated with the ACL or a particular role (i.e. "counts as" relationships, logic or causal relationships).
- *ACL* is the name of the communication language i.e. FIPA's ACL, KQML.
- $R_t = \{r_{1,t}, r_{2,t}, \ldots, r_{n,t}\}$ is a non-empty set of role names.
- S_t is the set of states of affairs that are true at time t.
- $SubSoc_t \in N$ is a set of identifiers for each sub-society of the current society *soc*.
- $f: (Ag_t \times T) \rightarrow \wp R_t \setminus \{\emptyset\}$, where T represents the set of time points. This function labels at time t each agent in the society with at least one role. Time is an input parameter in f in order to be able to

[4] We follow (Jones and Sergot 1996) on the use of the deontic operators O and P for the representation of obligations and permissions respectively.

[5] Intuitively, this definition should not allow cycles, i.e. agent A owns agent B which owns A. Therefore, we aim to provide a formalism to prevent such phenomena.

identify agents that change roles, e.g. *f(agent_0,1) = lecturer* and *f(agent_0,2) = professor*.

- g: $(Ag_t \times T) \rightarrow Owners_t$, where $Owners_t$ is a non-empty set of institutions and individual agents that have representatives in the society. This function maps at time t each agent in the society to its owner.
- h: $SubSoc_t \rightarrow Societies_t$ where $Societies_t$ is the set of all (sub-)society models. This function maps sub-society identifiers to the sub-societies, i.e. given the identifier *id*, it returns the $\Sigma_{id,t}$ model of the sub-society.
- k: $(ACL \times T) \rightarrow \wp acl\_constraints_t$. This function takes as input the ACL name and produces the constraints that are associated with that ACL.
- l: $(R_t \times T) \rightarrow \wp role\_preconditions_t \times \wp role\_constraints_t$ where $role\_preconditions_t$ is the set of the preconditions of the social roles. This function maps a role name to the set of constraints and preconditions that are associated with that role.

Time is taken to be composed of points and is assumed to be finite, discrete and linear. Temporal intervals are defined as pairs of points. The next time point of t, is $t + 1$. The components of the $\Sigma_{soc,t}$ model may change over time, i.e. society constraints may change etc. However, we assume that communication in the society is performed with the use of a single ACL. The specifications of this ACL may change over time.

3.2 Semantics of the Society Model

We presented our preliminary attempt to produce a formal model of agent societies. We mainly concentrated on identifying the principal components of agent societies and the interplay of these components. It has been suggested that (Wooldridge 2000) theories of agency should be computationally grounded if such theories are to be treated as specifications for systems and feasible to practically implement. We have done little work towards that direction. Therefore, we are currently investigating ways to give grounded semantics to our model.

One example of a grounded formalism for reasoning about agent systems is a semantic structure called Interpreted Systems (Fagin et al. 1995). In this context, a distributed system is described by identifying the states in which the agents and the environment are into. Furthermore, knowledge is ascribed to each agent from an external point of view. One way of giving grounded semantics to our society model can be in terms of Interpreted Systems and the extensions that have been proposed for this semantic structure. This issue will be investigated in further research.

4. ACL Semantics

Expressed mental attitudes. Our formal model is concerned with externally observable phenomena. For communication we introduce the notion of publicly expressed mental attitudes and thus separate them from personal internal mental attitudes. These need not be identical, agents may not be sincere. An agent expressing a desire to have an action performed without really desiring that action to be performed, may be testing the willingness of another agent to comply for example. This means that an agent does not need to hold a mental attitude as a precondition to expressing it. Thus meanings are specified from a public perspective (Singh 1998), rather than the private perspective of a single individual whose personal inferences are subjective.

Speech act. A speech act is a message and is a 6-tuple:

$$Speech\text{-}Act = Name \times Name \times perf \times content \times cid \times seq$$

Where *Name* is the domain of agent names (for sender and receiver) and the remaining parameters are performative, content, conversation identifier and sequence number within this conversation.

Partial social states. A *partial* social state stores public information including expressed mental attitudes and control variables (to control the flow of conversation in a protocol). Within this partial social state[6] is the state of *persistent normative relations* (dealing with long term relationships[7] that arise from past interactions) and the *conversation states* (dealing with all public information relating to the current conversations). A conversation state includes, among other things, variables and propositions describing the conversation and the history of speech acts in that conversation.

$$P\text{-}State = Persistent\ Soc\text{-}Rel \times Conv\text{-}Array$$

The *Conv-Array* maps a conversation identifier onto a conversation state:

$$Conv\text{-}Array = cid \rightarrow Conversation\text{-}State$$

Conversation identifiers are necessary to enable an agent to distinguish between different conversations when in it is involved in more than one at the same time.

ACL specification. The ACL specification allows for context dependence in ACL semantics. In particular the meaning of a speech act may depend on the protocol currently in use. Therefore the ACL specification has three parts: (1) the *Converse Function* gives norms (i.e. permissions and obligations) for subsequent speech acts based on the current conversation state; (2) the *Protocol Semantics* gives the meanings of speech acts in the context of the current protocol and (3) the *Speech-Act Semantics* give the protocol independent elements of meaning. For example, announcing may

[6] Each partial social state is a subset of the social state. A complete social state may include propositions that are not associated with communication.

[7] For example, after the end of an auction, the winning bidder is obliged to pay the auctioneer the specified amount. As long as it is not ‘satisfied’, this obligation will hold in the partial social state of subsequent interactions.

have a protocol independent meaning which makes some fact public, but it may also have additional meaning which varies depending on whether it is used in the context of an auction or a negotiation. Protocol specific meanings may override the speech act semantics since the protocol semantics are applied after the speech act semantics. The converse function is applied last. These three comprise the ACL specification which is given a semantics with the *language function* **L**.

Semantic definition. Each message is treated as a declarative statement that is given a procedural interpretation by a denotational semantics (Guerin and Pitt 2000). This semantics defines a speech act's semantics as a function between partial states.

As shown in Figure 1, the ACL specification is input to the denotational semantic functions. The *language function* **L** maps this specification onto a function from speech act to a (partial) state change.

$$\mathbf{L}: \mathit{ACLspecification} \rightarrow \mathit{Speech\text{-}Act} \rightarrow \mathit{P\text{-}State} \rightarrow \mathit{P\text{-}State}$$

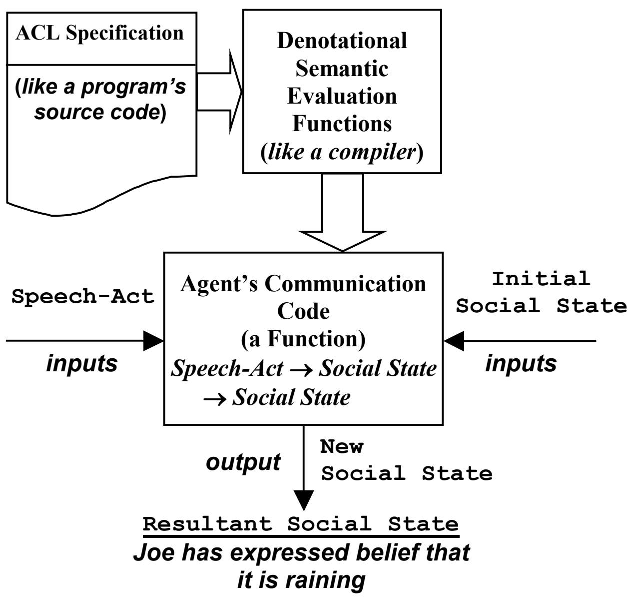

Figure 1. Communication Model

One can consider the ACL specification to be like the source code of a program, this is compiled with the denotational semantic functions (**L**) to produce the agent's running code which takes as input an initial partial social state and a new speech act, and produces as output the resulting partial social state. An agent must process any new communications in each cycle, including any it has just sent itself since both sender and receiver change conversation state.

Note that the conversation variables and propositions control the conversation, although they are labelled with beliefs, desires and intentions, they need not have any direct relationship with the internal states of the participants. The use of intentional labels makes the state description more intuitive for the agent designer.

5. Artificial Simulation

The *Alfebiite Animator* (Kamara et al. 2000) is a multi-agent testbed that provides a context for the simulation of physical agent environments[8]. The *Alfebiite Animator* (hereafter referred to as *"Animator"*) can be seen as an extension and refinement of the *Medlar Animato*r (Cunningham et al. 1996). The *Animator* has a *DAI* architecture, with a central server acting as host to multiple agent processes. The central server *(Reality)* maintains the physical environment, determining the outcome of agent physical actions. Each agent action is expressed through communication between the relevant agent process and Reality. To perform an action, for example, an agent sends an appropriately formatted message (complete with relevant parameters) to Reality and suspends until a reply is received (a time-out option exists). Reality, running asynchronously, reads the message and determines whether or not such an action translates into a valid action (given the current state of both agent and physical environment). In the case that it does not, an error message is returned. A valid physical action request, however, has the following consequence: Reality updates the environment to reflect successful performance of this action and then sends a positive acknowledgement to the agent complete with information about the effect of the action on the agent's own state. This basic request-reply pattern characterises all agent-environment interactions.

It can be seen at once that the Reality process functions not only as a communications medium between agent and environment, but that it also embodies the rules and constraints forming the physical characteristics of the environment. This allows the possibility of extended intervention in action requests, to simulate various effects such as permanent prohibition and distortion (of an agent's perceptions). In addition, the Reality process is able to monitor the activities of agents, for subsequent evaluation and analysis. The synchronisation mechanism underlying the Animator's architecture prevent invalid simulation configurations from arising, but agents can still attempt to perform 'wrong' or 'inappropriate' physical actions, which are duly recorded. It is important to note that the Reality process does not (directly) serve as a communications medium between agents themselves. Details of the communications platform are given in the following section.

The Animator, has been designed with a relatively generic interface in order to support variation in the type of scenario modelled as well as to facilitate extensibility. The key elements in defining a scenario

[8] The meaning of the term 'physical environment' is application-specific. An example of such an environment is a 2-dimensional grid. Similarly, we define the meaning of 'physical actions'. In a grid application a physical action would be a 'move' action.

are the allowable interactions; in the testbed, these translate into the message formats used and expected in communication between agent and Reality. An additional aspect is the use of a graphical display for the purpose of monitoring ongoing simulations. Such depictions are almost always scenario dependent and only make sense if they accurately reflect the intuition of the associated agent actions and environment. Figure 3 illustrates a graphical configuration for a scenario currently being modelled with the Animator. The graphical extensions are a server-side feature, serving to enhance the testbed as a monitoring and evaluation tool. A graphical interface is not necessary for client-side operation: it is sufficient for each client process to be able to communicate according to the protocol established by the scenario and enforced by Reality. This abstraction in communications allows a wide variety in the types of agents that may participate in the simulation.

6. The Integrated Animator

The Animator, as described in the previous section, is limited in the sense that it represents and deals only with the physical aspects of the simulated societies. However, we aim to produce analyses of the physical as well as the social aspect of the simulated multi-agent systems. Therefore, we have integrated the Animator with a prototype implementation of the society model (Artikis et al. 2001). We aim to further extend the functionality of the Animator. We propose a specification of the integration of the Alfebiite Animator with (1) a prototype implementation of a module that compiles the acl semantics and produces partial social states and (2) a communication platform that provides a medium for inter-agent communication and monitoring (Figure 2). The acl semantics module, given as input the communicative acts of the agents, calculates and updates each partial social state. The inputs of the society model (the $\Sigma_{soc,t}$ model) module are all the externally visible events[9] and the updated partial social states (provided by the acl semantics module).

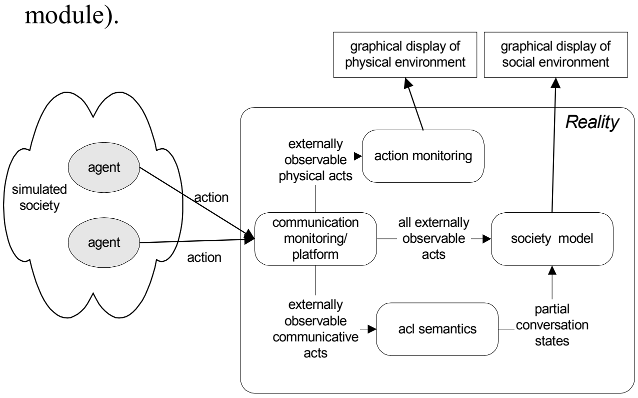

Figure 2. The Integrated Socio-cognitive Animator.

[9] All externally observable actions (physical, communicative or other) performed by each agent and time-out events.

The society module of Reality compiles the social structure of the simulated environment (part of this structure is the *complete* social state) not only for the benefit of an external observer; members of that society also have restricted access to the model. In particular, the $\Sigma_{soc,t}$ module can handle requests/queries from agents about the social state and the remaining components of the society model. However, each agent's view is qualified by its relative position in the social environment, as well as the (sub-) societies that it is a member of.

The components of the Reality (Figure 2) share a global clock (which is not necessarily visible to the agents) based on which they update the information they provide to external observers and to members of the simulated societies.

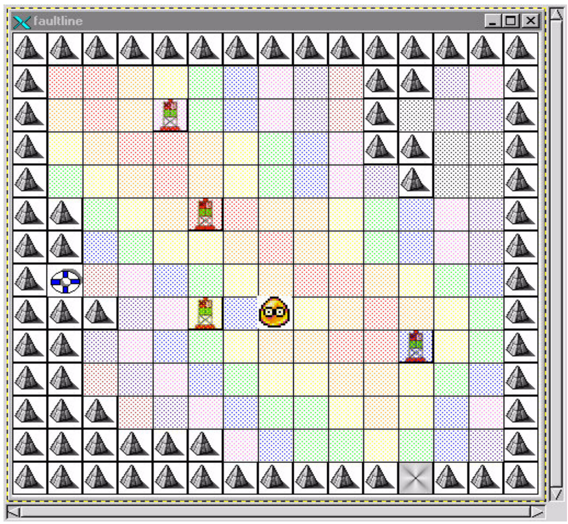

Figure 3. The GUI of a physical environment

Figure 3 demonstrates a snapshot of the GUI of the physical environment of a simulated society. The environment of this society is divided into a series of locations; one such location is featured in Figure 3. As can be seen, the basic character of the environment is a two-dimensional grid. Agents and objects are located at co-ordinates within the grid. In the case of the modelled scenario (one of oil exploration), objects include insurmountable obstacles (the "pyramids" in the screen-shot) ports (the buoy icon) and gates that allow movement between individual locations. The majority of the co-ordinates featured in the screen-shot have different colouring: this reflects the distribution of oil within the location, with each colour representing a specific oil content value. Such information can be made available to external observers but can be withheld from agents situated within the scenario. The graphical representation featured above is based upon the central model of the environment maintained by Reality.

Figure 4 illustrates a snapshot of the GUI of the social environment of a simulated marriage service. This snapshot shows the social constraints that relate an agent occupying the role of the priest with the agents that occupy the roles of the groom, bride and witness at the first time point of the simulation (i.e. $t=1$). For example, the priest is related with the groom, the bride and the witness due to constraint d2, because (according to d2) the priest is obliged to perform the ritual if and only if the groom and bride agree to get married and the witness confirms that. So, the meaning of this relationship lies in the fact that a particular action of the bride, groom and witness creates an obligation for the priest (see (Artikis et al. 2001) for more details on the animation of the marriage service protocol).

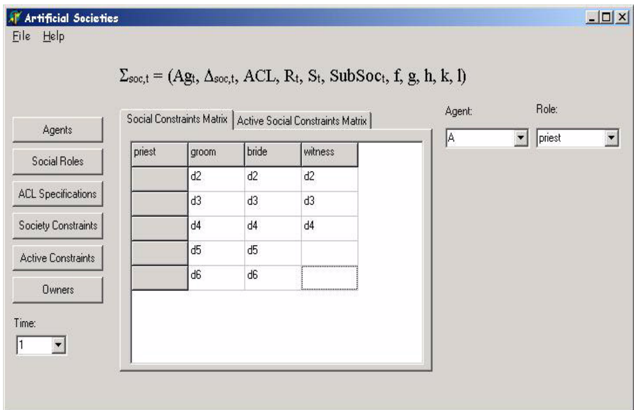

Figure 4. The GUI of a Social Environment.

7. Application Domain

In this section we present an e-commerce (producer/consumer) scenario and animate an agent society (that was formed in order to execute a trading protocol in the context of such a scenario) with the use of our theoretical framework. This animation illustrates the functionality of the proposed integrated Animator.

7.1 The Producer-Consumer Scenario

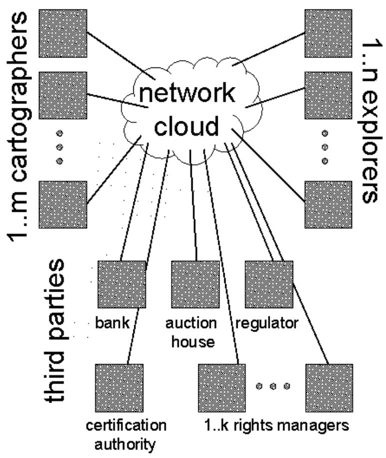

Figure 5. The Explorers/Cartographers Scenario.

In this *producer/consumer* scenario (see (Pitt et al., 2001) for a more detailed description of this scenario) explorer agents (producers) sell information to cartographers (consumers). The information commodity in question is geophysical in nature, with explorer agents mapping out the distribution of oil in their environment. The geophysical data is of variable quality; it is in the explorers' interests to find the best possible oil "plots", as these will fetch the best price on the market. It is in the cartographers' interests to get the best quality exploration data as these have the greatest intrinsic value. Trading is, of course, competitive, with individual explorers submitting their data for auction and cartographers initiating contracts for specific exploration. The scenario also features several authorities and institutes that facilitate and constrain interaction between the two main agent classes. This collection of agents, authorities, roles, regulations and interactions constitutes a society (see Figure 5).

7.2 Animation of the CNP in the Producer-Consumer Scenario

Cartographers can use the contract-net protocol to contract a particular explorer to search a certain region (see Figure 6). During the execution of the contract-net the behaviour of the participating cartographer and explorers is regulated by a set of social norms and conventions. For example, it is a convention of the contract-net sub-society that when an explorer-bidder submits a bid then this explorer expresses the desire to perform the task. The end of a successful (i.e. the task has been awarded to some bidder) contract-net results in the creation of some further norms and conventions. There exists a contract between the cartographer and the winner-explorer which defines a set of norms, i.e. the winner is obliged to perform the task and after the task has been performed, the contractor is obliged to pay the winner the agreed amount.

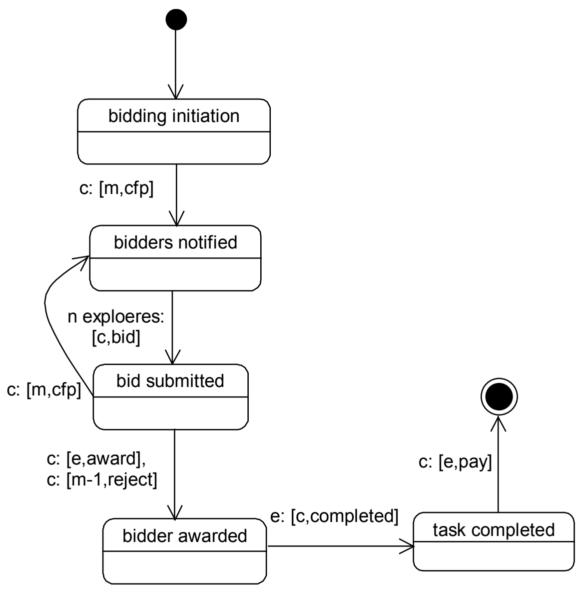

Figure 6. The Statechart Diagram of the CNP

Due to space limitations and for clarity reasons, we assume a simplistic and ‘correct’ execution of the CNP

(Figure 6), in the sense that we do not deal with 'errors' (e.g. a cartographer is informed that the awarded task is completed but does not pay for it). Furthermore, we do not address the issue of time (e.g. we do not deal with time-out events). Table 1 demonstrates a set of constraints of a simplistic specification of the CNP, while Table 2 illustrates the textual descriptions of these constraints. These constraints are included in the $\Delta_{cnp,t}$ set of our society $\Sigma_{cnp,t}$ model.

Table 1. Constraints on the CNP

d_1	$D_s(E_e bid(e,c,t) \wedge E_c award(bid(e,c,t)) \rightarrow OE_{CNP} Contract(c,e,t,price))$
d_2	$D_s(Contract(c,e,t,price) \rightarrow OE_e Task(t) \wedge OE_e completed(e,t,result))$
d_3	$D_s(Contract(c,e,t,price) \wedge E_e completed(e,c,t,result) \rightarrow OE_c pay(c,e,t))$

We animate the execution of the CNP (Table 3) by a group of 3 agents; two explorers e_1 and e_2 and a cartographer c. A sketch of this execution is the following: c performs a call for proposals (i.e. *cfp* speech act) and e_1 and e_2 submit their bids. Then the cartographer performs a new *cfp* (having modified the task description) and the two explorers submit their new bids. The cartographer awards the bid of e_1 and rejects the bid of e_2. e_1 performs the awarded task and c pays him the specified price.

Table 2. Textual Description of the Constraints of the CNP

d_1	If an explorer *e* bids for the task described by *t* and the cartographer *c* awards *e* with *t* then this counts in the society *CNP* as that there is a contract between *c* and *e* concerning *t* and *price*.
d_2	If there is a contract between the cartographer *c* and an explorer *e* concerning task *t* and *price* then *e* is obliged to bring about *Task* (i.e. do the task described by *t*) and inform *e* of the completion of *t*.
d_3	If there is a contract between the cartographer *c* and an explorer *e* concerning task *t* and *price*, and *e* informed of the completion of *t* then *c* is obliged to pay for *t*.

We represent the events that occur at each time step and the way these events modify the partial and the complete social state. As already mentioned in section 4, the components of the partial social state are the conversation state (which in turn is decomposed into variables and propositions that describe the conversation), the persistent normative relations and the history of the conversation. At each time step the set of propositions of the conversation state includes the social norms that are associated with the current events and the expressed mental attitudes that correspond to the currently performed speech acts (see section 4). For example, at time $t=0$ the cartographer performs the *cfp* speech act, thus he publicly expresses the desire that one of the explorers will bring about the task described by t_1. We represent publicly expressed mental attitudes as follows:

$$X <B|D|I>_{agent}<proposition>$$

B,D and I are the intentional labels for beliefs, desires and intentions respectively; the letter X that precedes the mental attitude states the fact that this mental attitude is publicly expressed.

At time $t=4$ the cartographer sends an *award* speech act to e_1 and a *reject* one to e_2. Thus the set of propositions (at this time step) includes, among other things, the cartographer's expressed desire that e_1 performs the task described by t_4 (this mental attitude corresponds to the *award* speech act) and the fact that in the CNP society, c and e_1 are bound in contract concerning t_4 and *price* (this fact corresponds to social constraint d_1).

The partial social state is a subset of the complete social state, therefore, all information that is in the partial social state is also included in the complete one. However, for clarity reasons we do not represent that in Table 3. We only show information in the complete social state column that exists only in the complete social state and not in the partial one. For example, at time $t=5$, e_1 brings it about that he performs the task described by t_4. This is not a communicative act so it does not modify the partial social state. However, this fact (i.e. that e_1 performed the task) is included in the complete social state where we store information about all acts.

The transition from one social state to the next is governed by the function that was outlined in section 3 (the fact that we augmented the complete social state with expressed mental attitudes does not affect the functionality of the *statechange* function). Based on that transition function, states of affairs that are included in the current state will also be included the subsequent one, unless these are retracted by the 'activation' of institutional constraints. However, for clarity reasons, in Table 3 we only represent information that is acquired by the current events.

The set of persistent normative relations (as shown in Table 3) is empty throughout the execution of the CNP. This set is empty because only one protocol takes place during the animation of the society (i.e. there cannot exist relationships from previous protocols) and all obligations are 'satisfied' before the end of that protocol. Due to the fact that we animate only a single protocol, we have not added a conversation identifier as a parameter in the exchanging speech acts and we therefore do not represent in Table 3 the conversation array.

It is important to note that a complete animation of the society would include a representation of all the components (i.e. social roles, owners, mapping of agents to their roles etc.) of the $\Sigma_{CNP,t}$ model at each time step. However, even for this simple execution of the CNP, space limitations make such representation in this paper infeasible.

Table 3. Animation of the CNP

Time	Event	Social State			
		partial social state			complete social state
		conversation state		persistent normative relations	
		variables	propositions		
0	$E_c cfp(c,\{e_1, e_2\},t_1,cnp)$	task_description=t_1 t_1={ price:40, task:map(X)} manager=c bidders={e_1, e_2} protocol=cnp	$E_c cfp(c,\{e_1, e_2\},t_1,cnp)$ $\forall ag \in \{e_1, e_2\}\ XD_c E_{ag} Task(t_1)$		
1	$E_{e1} bid(e_1,c,t_2) \wedge E_{e2} bid(e_2,c,t_3)$	t_2={ price:50, task:map(X)} t_3={ price:60, task:map(X)}	$E_{e1} bid(e_1,c,t_2)$ $E_{e2} bid(e_2,c,t_3)$ $XD_{e1} E_{e1} Task(t_2)$ $XD_{e2} E_{e2} Task(t_3)$		
2	$E_c cfp(c,\{e_1, e_2\},t_4,cnp)$	t_4={ price:45, task:map(X)}	$E_c cfp(c,\{e_1, e_2\},t_4,cnp)$ $XD_c E_{ag} Task(t_4)$ $\forall ag \in \{e_1, e_2\}$		
3	$E_{e1} bid(e_1,c,t_4) \wedge E_{e2} bid(e_2,c,t_5)$	t_4={ price:45, task:map(X)} t_5={ price:47, task:map(X)}	$E_{e1} bid(e_1,c,t_4)$ $E_{e2} bid(e_2,c,t_5)$ $XD_{e1} E_{e1} Task(t_4)$ $XD_{e2} E_{e2} Task(t_5)$		
4	$E_c award(c,e_1,bid(e_1,c,t_4)) \wedge$ $E_c reject(c,e_2,bid(e_2,c,t_5))$	t_4={ price:45, task:map(X)} winner= e_1	$E_c award(c,e_1,bid(e_1,c,t_4))$ $E_c reject(c,e_2,bid(e_2,c,t_5))$ $XD_c E_{e1} Task(t_4)$ $OE_{CNP} Contract(c,e_1,t_4,price))$		
5	$E_{e1} Task(t_4)$				$E_{e1} Task(t_4)$
6	$E_{e1} completed(e_1,c,t_4,result)$	result= map_of(X)	$E_{e1} completed(e_1,c,t_4,result)$ $XB_{e1} E_{e1} Task(t_4)$ $OE\ _c pay(c,e_1,price)$		
7	$E_c pay(c,e_1,price)$	price=45	$E_c pay(c,e_1,price)$ $XB_c E_{e1} Task(t_4)$		

8. Summary

8.1 Related Work

There exist several platforms specially built for the simulation of e-commerce applications. The Fishmarket auction house FM97.6 (Rodríguez-Aguilar et al. 1998) is a Java-based platform supporting the specification and evaluation of experimental trading scenarios (or 'tournaments'). To trade, buyers and sellers must adhere to the conventions governing the Fishmarket. There are ontological, social and individual aspects to the conventions; together, these characterise the institution underlying the simulation. As well as overseeing client agent behaviour, an institutor mechanism establishes the architecturally neutral aspect of the testbed. The system has a centralised control architecture, with simulated registration, auction and financial bodies providing conduits for agent interaction. While such an arrangement features implicit trust (participating agents relying upon the integrity of the controlling institutions), it does not support a dynamic trust model. Furthermore, there is a lack of formalisation of normative relations as well as an explicit representation of them during simulations.

The *Multi AGent Negotiation Testbed*, or MAGNET (Collins *et al.* 1998) offers facilities for modelling transaction protocols used by agents in simulated markets. MAGNET focuses on the machinery underlying electronic commerce - such as bidding, contracting and negotiation through intermediaries - rather than socio-cognitive aspects.

Griss and Letsinger (2000) describe the use of the Zeus multi-agent framework (Nwana et al. 1999) to model agent-mediated e-commerce, with agents taking on the roles of buyers, sellers, brokers and service providers. Their main concern is with the economics of such situations and whether market stability can be achieved. Issues of trust are also considered, insofar as they affect the market performance and relationships of agents (as in Kasbah (Chavez and Maes, 1996)) - for example, the effect of guarantees and insurance upon agent behaviour in collaborative scenarios. The model also extends to semi-open markets and identifies the role of regulation in such configurations. However, concepts like those of software (agent) ownership and the legal implications (which are key aspects of the integrated animator) are not yet directly addressed.

8.2 Conclusions

Agents' norm-governed behaviour and interactions facilitate the stability of artificial societies. We propose a new formal model of open artificial societies that focuses on externally observable states of affairs, and represents phenomena such as social constraints, social norms (obligations and permissions), communication language, social structure, ownership and society dynamics. This model investigates and enables regulation of the behaviour of the society members over time.

We propose a feasible specification for the integration of the society model with a multi-agent Animator to automate the examination of the agents' behaviour in a social heterogeneous context. Our attempt to represent the evolution of a trading community revealed that such an automation of the animation of artificial societies will facilitate the process of launching real open artificial societies.

8.3 Future Work

We are currently investigating ways to give grounded semantics to our society model. One of these ways can be the extension of the formalism of Interpreted Systems in order to provide more expressive ways to reason about the social aspects of a multi-agent system.

More steps need to be taken in order to specify a formal model of open agent societies. One of them is to produce a language for representing all the necessary concepts, such as social norms (e.g. rights), communication, expressed mental attitudes, social roles, actions, temporal aspects etc. Possible re-structuring of the society model may also be necessary. It can be argued that the constraints that stem from the ACL specifications and the remaining social constraints should be separated in two different sets. Finally, we aim to implement the specification of the integration of the Animator with the theoretical framework and evaluate this implementation in the context of e-commerce applications.

Acknowledgements

This work has been undertaken in the context of the EU-funded ALFEBIITE Project (IST-1999-10298), and this support is gratefully acknowledged.

References

Artikis A. and Pitt J. A Formal Model of Open Agent Societies. *Proc. of AA 2001*.

Artikis A., Kamara L. and Pitt J. Towards an Open Society Model and Animation. *Proceedings of Agent-Based Simulation II workshop*, Passau, 2001.

Chavez A. and Maes P. Kasbah: An Agent Marketplace for Buying and Selling Goods. *Proc. of 1st Conference on Practical Application of Intelligent Agents and MA Technology*, 75–90, 1996.

Collins J., Tsvetovat M. Mobasher B. and Gini M. MAGNET: A Multi-Agent Contracting System for Plan Execution. *Proc. of the AI and Manufacturing: State of the Art and State of Practice workshop*, AAAI Press, 63-68, 1998.

Cunningham J., Kamara L. and Pitt J. A Test-bed for Animating MAS. *The Practical Application of Intelligent Agents and MA Technology*, 1996.

Dellarocas C., Klein M. and Rodriguez-Aguilar J.A. An Exception-handling Architecture for Open Electronic Marketplaces of Contract Net Software Agents. *Proc. of the 2nd ACM Conference on E-Commerce,* Minneapolis, 2000.

Fagin R., Halpern J., Moses Y. and Vardi M. Reasoning about Knowledge. *MIT Press,* 1995.

Griss, M. and Letsinger, R. Games at Work - Agent-Mediated E-Commerce Simulation. *Proc. of the Agents in Industry Workshop*, AA 2000.

Guerin F. and Pitt J. Denotational Semantics for Agent Communication Languages. *Proc. of AA 2001*.

Jones, A., and Sergot, M. On the Characterisation of Law and Computer Systems: The Normative Systems Perspective. *Ch. 12 in Deontic Logic in Computer Science: Normative System Specification*, eds. Meyer and Wieringa, 1993.

Jones A., and Sergot M. A Formal Characterisation of Institutionalised Power. *Journal of IGPL* 4(3):429-445, 1996.

Kamara L., Neville B. and Pitt J. A Prototype Socio-Cognitive Simulator. Alfebiite, deliverable 1, number ist-1999-10298/d1, 164–180, 2000.

Nwana, H. Nduma, D., Lee, L. and Collis, J. ZEUS: a toolkit for building distributed multi-agent systems. *Journal of AI,* Vol. 13, No. 1, 129-186, 1999.

Pitt J., Kamara L. and Artikis A. Interaction Patterns and Observable Commitments in a Multi-Agent Trading Scenario. *Proc. of AA 2001*, (to appear).

Rodriguez-Aguilar J., Martin F., Noriega P., Garcia P. and Sierra S. Towards a Test-bed for Trading Agents in Electronic Auction Makets." *AI Communications*, vol. 11(1), 5-19, 1998.

Santos, F., Jones, A., and Carmo J. Action Concepts for Describing Organised Interaction. *Proc. 30th HICSS*, vol. V, 1997.

Singh, M. Agent Communication Languages: Rethinking the Principles. *IEEE Computer*, vol.31, no.12, 1998.

Venkatraman, M. and Singh, M. Verifying Compliance with Commitment Protocols. *Journal of Autonomous Agents and MAS*, vol. 2(3), 217-236, 1999.

Witkowski, M., Artikis, A., and Pitt, J. Trust and Cooperation in a Trading Society of Objective-Trust Based Agents. *Proc. of the Trust, Deception and Fraud Workshop of AA 2000*, 127-136, 2000.

Wooldridge M. Computationally Grounded Theories of Agency. *Proc. of ICMAS 2000,* IEEE Press, 2000.