An Abductive Interpretation for Open Agent Societies

M. Alberti[1], M. Gavanelli[1], E. Lamma[1], P. Mello[2], and P. Torroni[2]

[1] Dipartimento di Ingegneria, Università degli Studi di Ferrara
Via Saragat, 1, 44100 Ferrara, Italy
{malberti,mgavanelli,elamma}@ing.unife.it
[2] DEIS, Università degli Studi di Bologna
Viale Risorgimento, 2, 40136 Bologna, Italy
{pmello,ptorroni}@deis.unibo.it

Abstract. The focus of this work is on the interactions among (possibly heterogeneous) agents that form an open society, and on the definition of a computational logic-based architecture for agent interaction. We propose a model where the society defines the allowed interaction protocols, which determine the "socially" allowed agent interaction patterns. The semantics of protocols can be defined by means of social integrity constraints. The main advantages of this approach are in the design of societies of agents, and in the possibility to detect undesirable behavior. In the paper, we present the model for societies ruled by protocols expressed as integrity constraints, and its declarative semantics. A sketch of the operational counterpart is also given.

1 Introduction

In the Multi-Agent Systems community, societies of agents and agent interactions have been widely studied, and several models have been proposed.

Agent Communication Languages (ACL) and Conversation Protocols (CP) are the traditional approaches to support interactions among software agents. The semantics of speech acts in ACL is customarily defined in terms of mental attitudes such as Beliefs, Desires and Intentions [1]. This approach has been criticized as inadequate for open environments [2] since agents cannot verify whether the private beliefs of other agents comply with speech act definitions without pre-established constraints on how agents are internally implemented. On the other hand, CP are static structures that define the sequences of utterances making a coherent conversation. This approach has been criticized for its lack of flexibility, i.e., the lack of compositional rules governing how protocols are extended or merged.

In this work, we define the semantics of protocols as integrity constraints over social events (e.g., communicative acts), which caters for heterogeneity and openness, since it makes no assumptions on the internal structure of the agents.

We call such integrity constraints "social integrity constraints" (IC_S). IC_S express "constraints" on the communication patterns of agents, and therefore determine "expected" communicative acts, on the basis of the history of

A. Cappelli and F. Turini (Eds.): AI*IA 2003, LNAI 2829, pp. 287–299, 2003.

social events. The semantics for the ACL can be given in a uniform way, as done in [3, 4]. The overall proposed model for society protocols and ACL semantics has been developed in the context of the IST-2001-32530 project of the IST programme of the European Commission, titled *A computational Logic Model for the Description, Analysis and Verification of Global and Open Societies of Heterogeneous Computees - SOCS* [5].

Building on previous work on abductive logic-based agents, we assimilate the society's knowledge to abductive logic programs: we define a notion of *expected* social events, and express them as abducible predicates, while using IC_S to constrain the "socially admissible" communication patterns of agents. The syntax of IC_S and of the society in general are those of an extended logic program, and the semantics is inspired to that of abductive proof procedures such as the IFF [6]. This could lead to different kinds of verification: static, dynamic, and based on outside observation of the agents communication exchanges.

2 Society and Protocols

An open agent society needs to be tolerant to partial information, by continuing to operate despite the incompleteness of the available information. Also, the model of such a society should cope with the fact that information on agent interactions becomes available bit by bit over time. In our model, the society is time by time aware of social events that dynamically happen in the social environment. Moreover, the society can generate a set of "expected social events" that are not yet available to it. This set reflects the "ideal" behavior of the society and its members. Such expectations can be used by the society to behave proactively: they could be made public in order to try and influence the behavior of the agents, towards an ideal behavior.

Indeed, the expectations of the society are adjusted when it acquires new knowledge from the environment on social events that were not previously available. In this perspective, the society should be able to deal with unexpected social events from the environment, which violate the previous expectations.

This can be the case in an open environment where "regimentation" [7] cannot be assumed. Unexpected events can raise mechanisms of recovery from violation (e.g., sanctions), without affecting the society.

The knowledge in a society is composed of three parts: *Social Organization Knowledge Base* (SOKB), *Social Environment Knowledge Base* (SEKB), and a set $\mathcal{IC}_S$ of Social Integrity Constraints. $\mathcal{IC}_S$ is composed of IC_S expressing what is expected to happen or is expected not to happen, given some history of events. For example, an IC_S could state that the manager of a resource should give an answer to whomever has made a request for that resource. IC_S can produce expectations on the future.

The SOKB defines structure and properties of the society, namely: goals, roles, and common knowledge and capabilities. SOKB can change from time to time. However, this knowledge can be seen as *static* since it describes the

organization of a society which changes more slowly than the way the SEKB does.

The current instantiation of a society is described by the SEKB, which takes into account occurred events and expectations about social events. Thus, the SEKB dynamically evolves (it is updated much more frequently than the SOKB) and consists of:

- The set of happened events (history **HAP**), containing all the events in the form of atoms $\mathbf{H}(p)$ where p is a (ground) term. It represents the effects of agents' actions at the social level. During the evolution of the society via communication and interaction amongst agents, the set **HAP** can only dynamically grow, when new socially significant events happen.
- A set of expected events **EXP**, containing *expectations* on the future: events that should (but might not) happen in the future (indicated as atoms $\mathbf{E}(X)$, where X is a term), and events that should not (but might indeed) happen in the future (indicated as atoms $\mathbf{NE}(X)$).

Intuitively, an **H** atom represents a socially significant event that happened in the society, i.e., social events are mapped into **H** predicates.

E is a positive expectation about an event (the society expects the event to happen in order to fulfill its protocols) and **NE** is a negative expectation, (i.e., the society expects the event not to happen). Expectations can be seen as hypotheses of the society about the behavior of agents. Note the difference between $\neg\mathbf{E}(X)$ and $\mathbf{NE}(X)$. The first expresses the fact that the society does not have an expectation about the happening of event X (yet, if the event happens, no protocol will be violated), while the second expresses the fact that the society expects the event not to happen. By default (i.e., unless specified by **E** or **NE**), the society does not have expectations about events.

2.1 Syntax of the Social Organization Knowledge Base

We consider the $SOKB$ as a logic program; its syntax, in Backus-Naur form, is:

$$\begin{aligned} SOKB &::= [Clause]^\star \\ Clause &::= Atom \leftarrow Cond \\ Cond &::= ExtLiteral\ [\ \wedge ExtLiteral\ ]^\star \\ ExtLiteral &::= Literal \mid Expectation \mid Constraint \\ Expectation &::= [\neg]\mathbf{E}(Term\ [, T]) \mid [\neg]\mathbf{NE}(Term\ [, T]) \\ Literal &::= Atom \mid \neg Atom \mid true \end{aligned} \tag{1}$$

Atom and *Term* are intended as usual in Logic Programming [8]; *Constraint* is a constraint in the CLP sense [9].

T is an optional parameter representing the *time* in which the expectation holds. It is a variable with a finite domain that can be subject to constraints [9].

The variables are quantified as follows:

- Universally, if they appear only in literals of kind **NE** (and possibly constraints) with scope the singleton **NE**;
- Otherwise universally with scope the entire clause.

Both goal-directed and non goal-directed behavior for a society is supported. In our abductive understanding of the society expectation, a goal-directed society may expect a certain behavior from an agent in order to reach its goal. As an example, we can consider a society with the goal of selling items. In order to sell an item, the society might expect some "auctioneer" agent to open the auction. The goal of the society could be $\leftarrow sold(nail)$ and the society might have, in the SOKB, a rule of kind:

$$sold(Item) \leftarrow \mathbf{E}(tell(Auctioneer, Bidders, openauction(Item, Dialogue)), T)$$

which says that one way to sell an item is to have some agent telling a set of possible bidders that an auction is open for the item. The protocol of the auction (i.e., the way the auctioneer and the bidders interact) can be then specified in $\mathcal{IC}_S$.

2.2 Syntax of Social Integrity Constraints

The set $\mathcal{IC}_S$ relates socially significant (happened) events and expected events. Intuitively, IC_S in $\mathcal{IC}_S$ are (forward) rules used to produce *expectations* about the behavior of agents. They are used to check if an agent of the society behaves in a permissible way with respect to its "social" behavior. Their syntax, in BNF, is:

$$\begin{aligned}
\mathcal{IC}_S &::= [IC_S]^\star \\
IC_S &::= Body \rightarrow Head \\
Body &::= (Event|Expectation)\ [\wedge BodyLiteral]^\star \\
BodyLiteral &::= Event|Expectation|Literal|Constraint \\
Head &::= HeadDisjunct\ [\ \vee HeadDisjunct\ ]^\star|\bot \\
HeadDisjunct &::= Expectation\ [\ \wedge (Expectation|Constraint)]^\star \\
Expectation &::= [\neg]\mathbf{E}(Term\ [, T]) \mid [\neg]\mathbf{NE}(Term\ [, T]) \\
Event &::= [\neg]\mathbf{H}(Term\ [, T]) \\
Literal &::= Atom \mid \neg Atom \mid true
\end{aligned} \tag{2}$$

The rules of scope and quantification are as follows:

1. A variable must appear at least in an *Event* or in an *Expectation*.
2. The variables that appear both in the *Body* and in the *Head* are quantified universally with scope the entire IC_S.
3. The variables that appear only in the *Head* must appear in at least one *Expectation* (in eq. 2), have as scope the disjunct they belong to, and
   (a) if they appear in literals **E** or $\neg$**E** are quantified existentially;
   (b) otherwise they are quantified universally.
4. The variables that appear only in the *Body* have the *Body* as scope and
   (a) if they appear only in conjunctions of $\neg$**H**, **NE**, $\neg$**NE** or *Constraint*s are quantified universally;
   (b) otherwise are quantified existentially.
5. the quantifier $\forall$ has higher priority than $\exists$ (e.g., literals will be quantified $\exists\forall$ and not viceversa).

Example 1. If an agent X says "ask" to an agent Y during a conversation D, Y is expected to answer back either "yes" or "no", but not both "yes" and "no" (for the sake of simplicity, we do not report the constraints on time variables):

$$\begin{array}{l}\mathbf{H}(tell(X,Y,ask,D),T) \rightarrow \mathbf{E}(tell(Y,X,yes,D),T') \vee \ \mathbf{E}(tell(Y,X,no,D),T') \\ \mathbf{H}(tell(X,Y,yes,D),T) \rightarrow \mathbf{NE}(tell(X,Y,no,D),T') \\ \mathbf{H}(tell(X,Y,no,D),T) \ \rightarrow \mathbf{NE}(tell(X,Y,yes,D),T')\end{array}$$

For this example, we make scope and quantification of variables explicit:

$$\begin{array}{l}\forall X \forall Y \forall D(\exists T(\mathbf{H}(tell(X,Y,ask,D),T)) \rightarrow \exists T'(\mathbf{E}(tell(Y,X,yes,D),T')) \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \vee \ \exists T'(\mathbf{E}(tell(Y,X,no,D),T'))) \\ \forall X \forall Y \forall D(\exists T(\mathbf{H}(tell(X,Y,yes,D),T)) \rightarrow \forall T'(\mathbf{NE}(tell(X,Y,no,D),T'))) \\ \forall X \forall Y \forall D(\exists T(\mathbf{H}(tell(X,Y,no,D),T)) \ \rightarrow \forall T'(\mathbf{NE}(tell(X,Y,yes,D),T')))\end{array}$$

An agent may fulfill its expected behavior or not, so expectations may be fulfilled (by a history) or not. In this example, X telling "ask" to Y generates an expectation which can be fulfilled if Y answers back "yes" (Y behaves "properly"). Y violates the protocol, instead, if it says "no" after saying "yes", due to the second IC_S ("improper" behavior of Y).

At all times there can be alternative sets of expectations, as the previous example has shown. The alternative sets can be computed by a suitable proof procedure and provided to the agents (pro-actively) as a range of possibilities to comply to protocols (Sect. 4).

3 Declarative Semantics

In the following, we introduce a series of successively more refined declarative semantics. The various declarative semantics offer a range of options for different proof procedures, and are a ground basis to identify relevant properties of the society and its protocols.

Through this section, we consider the ground version of society's knowledge base and integrity constraints, and we do not consider CLP constraints.

We first introduce the concept of *admissible set of social expectations*. Intuitively, given a society and a set of events **HAP**, an admissible set of social expectations consists of a set of expectations about social events that are compatible with the SOKB, the set **HAP**, and the set $\mathcal{IC}_S$.

Definition 1. *Given a society and a set of events* **HAP***, an* admissible set *of social expectations* **EXP** *is a set of expectations such that:*

$$SOKB \cup \mathbf{HAP} \cup \mathbf{EXP} \models \mathcal{IC}_S \tag{3}$$

Many different sets of expectations are admissible, given $\mathcal{IC}_S$, SOKB, and **HAP**.

Example 2 (admissible set of expectations). Consider the following situation:

- $SOKB = \emptyset$
- $\mathbf{HAP} = \{\mathbf{H}(tell(thomas, yves, start)), \mathbf{H}(tell(david, yves, stop))\}$
- $\mathcal{IC}_S = \{\mathbf{H}(tell(X, Y, start)) \rightarrow \mathbf{E}(pass(Y)),$
  $\mathbf{H}(tell(X, Y, stop)) \rightarrow \mathbf{NE}(pass(Y))\}$

$\mathbf{EXP}_1 = \{\mathbf{E}(pass(yves)), \mathbf{NE}(pass(yves))\}$ is an admissible set of expectation, w.r.t. the $SOKB$, **HAP**, and $\mathcal{IC}_S$ above. Notice that any superset of $\mathbf{EXP}_1$ is also admissible. Instead, $\mathbf{EXP}_2 = \{\mathbf{E}(pass(yves))\}$ is not an admissible set of expectations, because $\mathbf{H}(tell(david, yves, stop)) \in \mathbf{HAP}$, $\mathbf{NE}(pass(yves)) \notin \mathbf{EXP}_2$, and thus the second integrity constraint in $\mathcal{IC}_S$ is violated.

Note that an admissible set of expectations could be self-contradictory (e.g., both $\mathbf{E}(p)$ and $\neg\mathbf{E}(p)$ may belong to an admissible set). More refined semantics can be given by identifying a subset of admissible expectation sets as intended semantics for a society. In particular, we are interested in those which are *coherent* and *consistent*:

Definition 2. *A set of social expectations* **EXP** *is* coherent *if and only if:*

$$\{\mathbf{E}(p), \mathbf{NE}(p)\} \not\subseteq \mathbf{EXP}$$

Example 3 (coherent set of expectations). Let us consider the situation presented in Example 2. $\mathbf{EXP}_2 = \{\mathbf{E}(pass(yves))\}$ is a coherent set of expectations (although it is not admissible w.r.t. the $SOKB$, **HAP**, and $\mathcal{IC}_S$). On the other hand, $\mathbf{EXP}_1 = \{\mathbf{E}(pass(yves)), \mathbf{NE}(pass(yves))\}$ is not a coherent set of expectations (although it is admissible).

Trivially, we are not interested in sets of social expectations that, at the same time, require that a particular event p should happen and should not happen.

Definition 3. *A set of social expectations* **EXP** *is* consistent *if and only if:*

$$\{\mathbf{E}(p), \neg\mathbf{E}(p)\} \not\subseteq \mathbf{EXP} \text{ and } \{\mathbf{NE}(p), \neg\mathbf{NE}(p)\} \not\subseteq \mathbf{EXP}$$

Example 4 (consistent set of expectations). Modification of Example 2:

- $SOKB = \emptyset$
- $\mathbf{HAP}= \{\mathbf{H}(tell(thomas, yves, start)), \mathbf{H}(tell(david, yves, stop))\}$
- $\mathcal{IC}_S = \{\mathbf{H}(tell(X, Y, start)) \rightarrow \mathbf{E}(pass(Y)),$
  $\mathbf{H}(tell(X, Y, stop)) \rightarrow \neg\mathbf{E}(pass(Y))\}$

The intuitive meaning of the second constraint in $\mathcal{IC}_S$ is: *If I tell you "stop" then one should not expect that you pass.*

$\mathbf{EXP}_2 = \{\mathbf{E}(pass(yves))\}$ is a consistent set of expectations, although it is not admissible, since $\mathbf{H}(tell(david, yves, stop)) \rightarrow \neg\mathbf{E}(pass(yves)) \in \mathcal{IC}_S$, $\mathbf{H}(tell(david, yves, stop)) \in \mathbf{HAP}$, and $\neg\mathbf{E}(pass(yves)) \notin \mathbf{EXP}_2$. $\mathbf{EXP}_3 = \{\mathbf{E}(pass(yves)), \neg\mathbf{E}(pass(yves))\}$ is instead an admissible set of expectations, but it is not consistent.

We are not interested in sets of social expectations that are intrinsically inconsistent, i.e., that at the same time, expect something and do not expect the same thing. When no coherent (and consistent) admissible expectation set exists, and therefore an incoherence (or inconsistency) arises, it means that the society has been modelled in a wrong way.

We would like to stress that we do not assume that expected events actually happen. This is in accordance with an *open* view for society where social expectations are just a suggestion for what should be done (or not done). It can be the case that in a situation an expectation is raised, but the expected event does not happen (this might lead to a violation, and possibly a sanction).

A further refined semantics is then given by identifying, among coherent and consistent admissible expectation sets, those which are *fulfilled* by a set of events happened in a society. This reflects the *ideal* behavior of a society.

Definition 4. *Given a society and a set of events* $\mathbf{HAP}$*, a coherent and consistent admissible set of social expectations* $\mathbf{EXP}$ *is* fulfilled *if and only if:*

$$\mathbf{HAP} \cup \mathbf{EXP} \models \{\mathbf{E}(p) \rightarrow \mathbf{H}(p)\} \cup \{\mathbf{NE}(p) \rightarrow \neg\mathbf{H}(p)\} \tag{4}$$

Example 5 (fulfilled set of expectations). Let us consider the following situation:

- $SOKB = \emptyset$
- $\mathbf{HAP}_1 = \{\mathbf{H}(tell(thomas, yves, start)), \mathbf{H}(pass(yves))\}$
- $\mathcal{IC}_S = \{\mathbf{H}(tell(X, Y, start)) \rightarrow \mathbf{E}(pass(Y))\}$

$\mathbf{EXP}_2 = \{\mathbf{E}(pass(yves))\}$ is a coherent, consistent and fulfilled admissible set of expectations, w.r.t $SOKB$, $\mathbf{HAP}_1$, and $\mathcal{IC}_S$. But if we consider a different history: $\mathbf{HAP}_2 = \{tell(thomas, yves, start)\}$, $\mathbf{EXP}_2$ is not a fulfilled expectations set w.r.t. $SOKB$, $\mathbf{HAP}_2$, and $\mathcal{IC}_S$ (still, it is admissible, coherent, and consistent).

Many different $\mathbf{EXP}$ sets are admissible with respect to social integrity constraints, the SOKB and the history $\mathbf{HAP}$. By Definition 4, we select, among them, those where the happened events cover all the events that should happen, and none of the events that should not happen.

Notice that such a fulfilled $\mathbf{EXP}$ set might not exist, even if a coherent and consistent admissible expectation set exists. The reason is the *violation* of the protocol: some agent did not behave as expected, and some action will be taken to recover from this violation.

Definition 5. *Given a society and a set* $\mathbf{HAP}$ *of events, if each coherent and consistent admissible set of expectations is not fulfilled (i.e., if* $\mathbf{E}(p) \rightarrow \mathbf{H}(p)$ *or* $\mathbf{NE}(p) \rightarrow \neg\mathbf{H}(p)$ *are violated), then we say that* $\mathbf{HAP}$ *produces a* violation.

Until now, we did not deal with a possible *goal* of the society. If we want to consider a goal-directed society, then we introduce the following definition.

Definition 6. *Given a society, a goal G and a set of events* **HAP**, *we say that G is* achievable *iff there exists a coherent and consistent admissible set of social expectations* **EXP**, *such that:*

$$SOKB \cup \mathbf{HAP} \cup \mathbf{EXP} \models G \tag{5}$$

Example 6 (achievable goal of a society). Let us consider the following situation:

- $SOKB = \{G_1 \leftarrow \mathbf{E}(pass(yves)),\ G_2 \leftarrow \mathbf{E}(pass(david))\}$
- $\mathbf{HAP} = \{\mathbf{H}(tell(yves, david, stop))\}$
- $\mathcal{IC}_S = \{\mathbf{H}(tell(X, Y, start)) \rightarrow \mathbf{E}(pass(Y)),$
  $\mathbf{H}(tell(X, Y, stop)) \rightarrow \mathbf{NE}(pass(Y))\}$

G_1 is an achievable goal w.r.t. $SOKB$, **HAP**, and $\mathcal{IC}_S$, thanks to the coherent and consistent admissible set of social expectations $\mathbf{EXP} = \{\mathbf{E}(pass(yves))\}$.

On the other hand, there exists no coherent and consistent admissible set of social expectations to make G_2 achievable, given history **HAP**.

Note that the notion of goal achievability does not guarantee that the goal is really achieved, since expectations may not be fulfilled, i.e., the corresponding events that should happen can be possibly not generated, and vice-versa.

Definition 7. *Given a society, a goal G and a set of events* **HAP**, *G is* achieved *iff there exists a fulfilled coherent and consistent admissible set of social expectations* **EXP** *such that Eq. 5 holds.*

Example 7 (achieved goal of a society). Let us consider the following situation:

- $SOKB = \{G_1 \leftarrow \mathbf{E}(pass(yves)),\ G_2 \leftarrow \mathbf{E}(pass(thomas))\}$
- $\mathbf{HAP} = \{tell(yves, david, stop), pass(yves)\}$
- $\mathcal{IC}_S = \{\mathbf{H}(tell(X, Y, start)) \rightarrow \mathbf{E}(pass(Y)),$
  $\mathbf{H}(tell(X, Y, stop)) \rightarrow \mathbf{NE}(pass(Y))\}$

G_1 is an achieved goal. G_2 is achievable, but it has not yet been achieved.

4 The Society Knowledge as an Abductive Logic Program

In our approach, agents autonomously perform some form of reasoning, while the society management infrastructure is devoted to ensuring that the agents do not collide with established rules and protocols.

By specifying protocols as IC_S, we can exploit an associated proof procedure to be used directly by a Society Infrastructure for verification, by using the intensional knowledge on constraints. Allowed paths can be inferred and not explicitly stated, avoiding over-constrained protocols.

We represent the knowledge available at social level as an Abductive Logic Program (ALP) [10], since such knowledge is generally incomplete. The idea of using abduction to model agent interaction is derived from a work by Kowalski

and Sadri on abductive agents [11], where the abducibles are produced within an agent cycle, and represent actions in the external world.

Events, expectations and society knowledge and protocols can be smoothly recovered into an abductive framework, so to exploit well-assessed proof-theoretic techniques in order to check the compliance of the overall computation with respect to the expected social behavior.

Using abduction to record expectations allows for two features. First, it enlarges the dynamic knowledge available at social level during the agents' own reasoning, through knowledge acquisition. Second, it is a way to make relevant knowledge available to all the agents in the society.

At the society level, knowledge can therefore be represented as the triple: $\langle KB, \mathcal{E}, IC \rangle$ where:

- KB is the knowledge base of the society. It includes the SOKB and happened events ($\mathbf{HAP} \subseteq SEKB$);
- $\mathcal{E}$ is a set of *abducible predicates*, standing for positive and negative expectations or their negation;
- IC is the set of Social Integrity Constraints, $\mathcal{IC}_S$.

Abduction captures relevant events (or hypotheses about future events), and a suitably extended abductive proof-procedure can be used for integrity constraint checking. Given a goal G at the society level (see also Section 3), then G is achieved when[1]:

$$KB \vdash_{\mathbf{EXP}} G \tag{6}$$

and $\mathbf{EXP} \subseteq \mathcal{E}$ is a (coherent and consistent) set of abduced atoms such that

$$KB \cup \mathbf{EXP} \vdash IC \tag{7}$$

$$KB \cup \mathbf{EXP} \vdash \{\mathbf{E}(X) \rightarrow \mathbf{H}(X)\} \cup \{\mathbf{NE}(X) \rightarrow \neg\mathbf{H}(X)\} \tag{8}$$

if this last condition is not verified, then a *violation* occurs (see Def. 5).

Notice that Eq. 7 is the operational counterpart of Eq. 3 and Eq. 8 is that of Eq. 4. If the society is not goal-directed (no goal is specified for it), Eq. 6 is always satisfied for any set of expectations and, therefore, only equations 7 and 8 are significant (as in the declarative semantics, Section 3).

In this section we gave an abductive interpretation of the social framework. A suitable proof procedure will have to be defined in order to efficiently deal with such a semantics in a dynamic settig. In particular, the compliance verification to the specified $\mathcal{IC}_S$ and fulfillment check should be incremental (in order to detect violations as soon as possible), and the operational phases of equations from 6 to 8 should be interleaved properly. Complexity will also be an issue to address.

[1] We do not commit at this stage to any particular semantics for the $\vdash$ symbol. Many semantics indeed could be given, such as for instance the classical SLDNF derivation as usual in Logic Programming. $KB \nvdash G$ is a shorthand for $not(KB \vdash G)$. The symbol $\vdash_\Delta$ stands for an abductive derivation with set of abduced atoms Δ.

The IFF proof procedure [6] or an extension of it could be a plausible candidate because (*i*) it employs integrity constraints with forward reasoning, like our IC_S, and (*ii*) it keeps updated a "frontier" of alternative derivations, that could be communicated to agents as a range of socially satisfactory behaviors.

5 Related Work

Considerable work has been devoted to studying the concepts of norms, commitment and social relations in the context of multi-agent systems [12], and to proposing architectures for developing agents with social awareness [13]. Our approach can be considered complementary to these efforts, since it is mainly focused on the definition of a society infrastructure based on computational logic to regulate and improve robustness of interaction in an open environment, where the internal architecture of the agents might be unknown.

We have advocated a *social* approach as in [2], where the semantics of agent interaction is defined in terms of the effects that it has at a social level. Following this approach, even if the agents mental state cannot be accessed, it is possible to verify whether communicating agents in a society comply to the protocols that regulate their interaction. A *protocol* specifies the "rules of encounter" governing a dialogue between agents. It will usually allow for several alternative utterances in every situation and the agent in question has to choose one according to its private *policy*. A good protocol will enable fruitful interaction in general. A good policy will benefit the agent using it. The protocol is *public*, while each agent's policy is *private*. Protocols are practically important because they may help to select the adequate answer to an incoming utterance, thus reducing the complexity of this task for an agent [14].

The idea of expected behavior can be considered related to *deontic logic* [15]; however, our claim is that we do not need the full power of the standard deontic logic, but only constraints on events that are expected to happen or not to happen. We do not use deontic operators, but instead we map them into predicates (**E** for positive and **NE** for negative expectations).

Our work is very close for the objective and methodology to the notable work on computational societies presented and developed in the context of the ALFEBIITE project [16], and the work by Singh [17] where a social semantics is exemplified by using a commitment-based approach. With this work we share the same view of an open society as that of [18].

Artikis et al. [18] present a theoretical framework for providing executable specifications of particular kinds of multi-agent systems, called open computational societies, and present a formal framework for specifying, animating and ultimately reasoning about and verifying the properties of open computational society: systems where the behavior of the members and their interactions cannot be predicted in advance. Differently from [18], we do not explicitly represent the institutional power of the members and the concept of valid action. Permitted are all social events that do not determine a violation, i.e., all events that are not explicitly forbidden are allowed, and this implements a sort of "open

world assumption" at a society level. Permission, when it needs to be explicitly expressed, is mapped into the negation of a negative expectation: $\neg\mathbf{NE}(\ldots)$.

The semantics of our model can be directly mapped in an abductive framework, where expectations can be confirmed (fulfilled) or disconfirmed (violated) by the history of the happened social events.

Sadri et al. [19] propose a framework for agent negotiation based on dialogue. The dialogue of agents is defined in a two-part setting as an ordered sequence of communication primitives. The generation of dialogues results from an abductive reasoning process taking place inside each agent during the *think* phase of its life cycle (the cycle being inspired by [11]). Our work shares the view of integrity constraints that provide new abducible atoms, but in our case the abducibles are *expectations* of the society about the future behavior of the agents, while in [19] they are used as communication primitives.

6 Conclusions

We have presented a computational-logic based approach for modelling interactions among (possibly heterogeneous) agents that form an open society. The allowed interaction protocols, which determine the "socially" allowed agent interaction patterns, are expressed as (social) integrity constraints. The main advantages of this approach are in the design of societies of agents, and in the possibility to detect undesirable behavior.

We have shown that the overall model can be mapped into an abductive logic programming framework, and presented a declarative semantics as clean extension of that for (extended) Logic Programming. This makes existing results for logic programming and monotonic reasoning re-usable in our context (see, for instance, [20]). Future work will be devoted to extend the IFF proof procedure [6], in order to make it applicable to our purposes.

We think that the proposed model addresses a basic aspect and a major engineering problem, paving the way for different kinds of verification: static, dynamic, and based on outside observation of the agents communication exchanges. Preliminary results on that have been presented in [21]. The full implementation of such verification is subject for future work.

The proposed model is independent from the agents' internals, and it is applicable to societies of heterogeneous agents. Nonetheless, if agents were aware of social expectations, they could plan and act appropriately in order to achieve them. Future work will be also devoted to studying the influence and integration of raised expectations within the agents' behavior cycle.

Finally, recovery from violation, i.e., deciding what to do when a violation occurs in a society, is also a fascinating future work subject.

Acknowledgements

This work is partially funded by the Information Society Technologies programme of the European Commission under the IST-2001-32530 project.

References

[1] Rao, A. S., Georgeff, M. P.: Modeling rational agents within a BDI-architecture. In Fikes, R., Sandewall, E., eds.: Proc. of KR&R-91, (Morgan Kaufmann) 473–484 287

[2] Singh, M.: Agent communication language: rethinking the principles. IEEE Computer (1998) 40–47 287, 296

[3] Alberti, M., Ciampolini, A., Gavanelli, M., Lamma, E., Mello, P., Torroni, P.: Logic based semantics for agent communication languages. In: Proc. of the Workshop of Formal Approaches to Multi-Agent Systems (FAMAS03), Warsaw (2003) 288

[4] Alberti, M., Ciampolini, A., Gavanelli, M., Lamma, E., Mello, P., Torroni, P.: A social ACL semantics by deontic constraints. In V.Marik, J.Muller, M.Pechoucek, eds.: Proc. of the 3rd International/Central and Eastern European Conference on Multi-Agent Systems, Prague, Czech Republic (2003) To appear in LNCS. 288

[5] SOCS: A Computational Logic Model for the Description, Analysis and Verification of Global and Open Societies of Heterogeneous Computees. IST–2001–32530. Web page (2001) `http://lia.deis.unibo.it/research/socs`. 288

[6] Fung, T. H., Kowalski, R. A.: The IFF proof procedure for abductive logic programming. Journal of Logic Programming **33** (1997) 151–165 288, 296, 297

[7] Krogh, C.: Obligations in multiagent systems. In Proc. of the 5th Scandinavian Conference on Artificial Intelligence, Trondheim, Norway, ISO Press (1995) 19–30 288

[8] Lloyd, J. W.: Foundations of Logic Programming. Springer-Verlag (1987) 289

[9] Jaffar, J., Maher, M.: Constraint logic programming: a survey. Journal of Logic Programming **19-20** (1994) 503–582 289

[10] Eshghi, K., Kowalski, R. A.: Abduction compared with Negation by Failure. In: Proc. of the 6th International Conference on Logic Programming. (1989) 294

[11] Kowalski, R. A., Sadri, F.: From logic programming to multi-agent systems. Annals of Mathematics and Artificial Intelligence (1999) 295, 297

[12] Conte, R., Falcone, R., Sartor, G.: Special issue on agents and norms. Artificial Intelligence and Law **1** (1999) 296

[13] Castelfranchi, C., Dignum, F., Jonker, C., Treur, J.: Deliberative normative agents: Principles and architecture. In: ATAL'99. Number 1757 in LNCS 364–378 296

[14] Endriss, U., Maudet, N., Sadri, F., Toni, F.: Protocol conformance for logic-based agents. In: Proc. of IJCAI 2003, (Morgan Kaufmann Publishers) 296

[15] Wright, G.: Deontic logic. Mind **60** (1951) 1–15 296

[16] ALFEBIITE: A Logical Framework for Ethical Behaviour between Infohabitants in the Information Trading Economy of the universal information ecosystem. IST–1999–10298. Web page (1999) `http://www.iis.ee.ic.ac.uk/alfebiite/`. 296

[17] Yolum, P., Singh, M.: Flexible protocol specification and execution: applying event calculus planning using commitments. In Castelfranchi, Lewis Johnson, eds.: Proc. of AAMAS'02, (ACM) 527–534 296

[18] Artikis, A., Pitt, J., Sergot, M.: Animated specifications of computational societies. In Castelfranchi, Lewis Johnson, eds.: Proc. of AAMAS'02, (ACM) 1053–1061 296

[19] Sadri, F., Toni, F., Torroni, P.: An abductive logic programming architecture for negotiating agents. In Greco, Leone, eds.: Proc. JELIA'02. Vol. 2424 of LNCS. 419–431 297

[20] Brogi, A., Lamma, E., Mancarella, P., Mello, P.: A unifying view for logic programming with non-monotonic reasoning. TCS **184** (1997) 1–59 297

[21] Alberti, M., Gavanelli, M., Lamma, E., Mello, P., Torroni, P.: Specification and verification of agent interactions using social integrity constraints. In: Workshop on Logic and Communication in Multi-Agent Systems, Eindhoven, Netherlands (2003) To appear. 297