A Social ACL Semantics by Deontic Constraints

Marco Alberti[1], Anna Ciampolini[2], Marco Gavanelli[1], Evelina Lamma[1], Paola Mello[2], and Paolo Torroni[2]

[1] Dipartimento di Ingegneria,
Università degli Studi di Ferrara
Via Saragat, 1 - 44100 Ferrara (Italy)
{malberti|mgavanelli|elamma}@ing.unife.it
[2] Dipartimento di Elettronica, Informatica e Sistemistica,
Università degli Studi di Bologna
Viale Risorgimento, 2 - 40136 Bologna (Italy)
{aciampolini|pmello|ptorroni}@deis.unibo.it

Abstract. In most proposals for multi-agent systems, an Agent Communication Language (ACL) is the formalism designed to express knowledge exchange among agents. However, a universally accepted standard for ACLs is still missing. Among the different approaches to the definition of ACL semantics, the *social* approach seems the most appropriate to express semantics of communication in open societies of autonomous and heterogeneous agents.
In this paper we propose a formalism (*deontic constraints*) to express social ACL semantics, which can be grounded on a computational logic framework, thus allowing automatic verification of compliance by means of appropriate proof procedures. We also show how several common communication performatives can be defined by means of deontic constraints.

1 Introduction

Agent communication is one of the key issues in multi-agent systems. Traditional interprocess communication mechanisms are usually viewed as too low-level to effectively represent knowledge exchange among agents: thus, most proposals of standards for multi-agent systems propose an Agent Communication Language (ACL for brevity hereafter) as a formalism that abstracts away from implementation issues and provides sufficient expressive power for knowledge exchange at a high level.

However, there is still no agreed upon standard for ACLs, and many different approaches can be found in literature.

Mentalistic approaches [1, 2] define ACL semantics in terms of agents' mental states. For instance, FIPA ACL [2] assumes a BDI (Belief, Desire, Intention) model for the agents, and relies upon it for defining the semantics of communicative acts in terms of Feasibility Preconditions (i.e., the conditions that have to be satisfied for the communicative act to be planned) and Rational Effects (i.e., the expected effect of the communicative act).

While the mentalistic approach is adequate to account for *motivation* of communicative acts from an internal viewpoint, it is not safe to rely on it in open environments, when it is impossible to verify the effect of communication on the internal state of agents, as explained by Singh [3].

The social approach [4] moves from criticism to the mentalistic approach, and defines ACL semantics in terms of the effects of the communicative acts on a society of agents. Following this approach, even if agents' mental state cannot be accessed, it is possible to verify whether communicating agents in a society comply to the social laws that regulate the interactions. Thus, the social approach appears more appropriate to define the semantics of communication in open societies of agents.

Significant social proposals are based on the concept of *commitments* [5, 6], computational objects which keep track of agents' obligations and can evolve depending on agents' (observable) communicative and non-communicative acts, and eventually be resolved to fulfillments or violations.

In this paper, we present a logic-based approach to the definition of an ACL semantics which follows the principles of the social approach. It defines the meaning of communicative acts not as their effect on the mental state of single agents (upon which no *a priori* assumptions are made) but as their (verifiable) social effect. Our semantics is based on the concept of *expectation* (which, as explained in Sect. 4, is broader than that of commitment as adopted in [5]), linked to *socially significant events* by *deontic integrity constraints*, which provide a declarative specification of the proper interaction behavior in a society of agents. This specification can be grounded on a computational logic framework [7], thus obtaining the advantage of automatic verification by an appropriate proof procedure.

With respect to previous work [7], in this paper we introduce the concept of *alternative* expectations, which allow for different sequences of events to be compliant with protocols and semantics of communication.

The paper is structured as follows. In Sect. 2 we introduce the concept of deontic constraint, and in Sect. 3 we show how it can be exploited to specify the social semantics of communicative acts; discussion and directions for future work follow.

2 Interaction among agents and deontic constraints

In this section, we briefly introduce our model of agent society, and focus on *deontic integrity constraints* (also *deontic constraints* or ic_Ds for brevity in the following) as a formalism to specify the desired behavior of interaction in a society of agents.

We consider interaction in an open society of autonomous and heterogeneous agents, and do not make any assumptions about the internal structure of individual agents. Deontic constraint are intended to constrain only the behavior of agents as required to ensure the desired operation of the society; thus, they regard *observable* and *socially significant* (i.e., effective over other agents or the environment[3]) events. We do not deal with modifications of the internal state of an agent that are not socially significant.

In the following, we assume that agents interact through a *social infrastructure*. This should be understood as a functional abstraction of a set of computational entities that provides a platform for agents' interactions. In particular, the social infrastructure is supposed to be aware of agents' socially significant actions and changes in the environment, and to be able to detect fulfillment or violations of expectations (explained later) by means of appropriate proof procedures.

A more detailed discussion of our model of society and of the functionalities of the social infrastructure can be found in [7].

Events can be produced by agents' actions or by modifications in the environment. We describe the content of events by atoms in a first order language. For instance, the atom

$$\mathit{request}(a_1, a_2, \mathit{give}(\mathit{nail}), \mathit{a\_dialog}, 10)$$

would represent a request issued at time 10, in the context of the dialog *a_dialog*, from agent a_1 to a_2 to *give* a *nail*.

We express the fact that an event has happened by the functor **H**. For instance, the event whose content is described above is represented as

$$\mathbf{H}(\mathit{request}(a_1, a_2, \mathit{give}(\mathit{nail}), \mathit{a\_dialog}, 10))$$

The desired behavior of a society of agents is expressed by *expectations*: events that should happen (*obligations*) and events that should not happen (*prohibitions*). Obligations and prohibitions are respectively expressed by the functors **O** and **ON**.

Deontic constraints relate events and expectations: intuitively, a deontic constraint expresses that, if some events have happened, then some other events are expected to (or not to) happen. Variables in an expectation can be constrained. We give here the syntax of deontic constraints, in BNF.

$$\begin{aligned}
ic_D &::= \chi \rightarrow \phi \\
\chi &::= \mathit{ExtendedLiteral}\ [\ \wedge \mathit{ExtendedLiteral}\ ]^\star \\
\phi &::= \mathit{OughtList}[:\ \mathit{ConstraintList}][\vee \mathit{OughtList}[:\ \mathit{ConstraintList}]]^\star \\
\mathit{ExtendedLiteral} &::= \mathit{EventAtom} \mid \mathit{Literal} \\
\mathit{DeonticAtom} &::= \mathbf{O}(\mathit{Atom}) \mid \mathbf{ON}(\mathit{Atom}) \\
\mathit{EventAtom} &::= \mathbf{H}(\mathit{Atom}) \\
\mathit{Literal} &::= \mathit{Atom} \mid \neg \mathit{Atom} \\
\mathit{OughtList} &::= \mathit{DeonticAtom}\ [\ \wedge \mathit{DeonticAtom}\ ]^\star
\end{aligned} \tag{1}$$

[3] By environment, we mean the setting in which the agent operate. In this work, we do not formalize the concept; however, we suppose that significant properties of the environment can be described by a first order language.

Atom represents an atom as intended in logic programming, and *ConstraintList* is a conjunction of constraints on the variables in *OughtList*. For instance,

$$\mathbf{O}(do(a_1, a_2, give(nail), a\_dialog, T)) : T \leq 10$$

means that, in the context of dialog *a_dialog*, agent a_1 is expected to *give* a *nail* to agent a_2 by time 10.

Expectations may be fulfilled or violated by events. For instance, the following event:

$$\mathbf{H}(do(a_1, a_2, give(nail), a\_dialog, 7))$$

would fulfill the above expectation. In case of violations, appropriate recovery procedures should be triggered so to bring the society back to a proper behavior.

A formal declarative semantics of deontic constraints is described in [7].

Compliance of events to deontic constraints can be achieved by means of appropriate proof procedures in a computational logic setting (as done, for a restricted version of deontic constraints, in [7]). Intuitively, deontic constraints can be used as production rules to generate the expectations in the head when the events in the body happen; these expectations can be fulfilled or violated by the history of events. Obviously, in order to operate in real world applications, such a proof procedure should operate incrementally, i.e., generate expectations and detect fulfillment and violation of them while new events happen.

3 Constraint-based ACL semantics

In this section, we show how social semantics of an ACL can be defined by means of deontic constraints. We consider communicative acts from a social perspective, i.e., we do not constrain the internal communicative model of agents (by which communicative acts are *motivated*), but instead link communicative acts to the expectations they generate. Social semantics of communicative acts is thus verifiable regardless of the internal structure of agents.

Throughout this section, we refer to the classification of primitives described in [5] (see Sect. 4), and define some primitives for each of the classes considered in [5], namely *assertives*, *commissives*, *directives* and *proposals*.

In the remainder, we describe a communicative act by an atom of this structure:

$$primitive\_name(Sender, Receiver, Content, Dialog, Time) \tag{2}$$

Assertive primitives: inform. Assertive primitives are used to exchange pieces of knowledge between agents, e.g. to communicate the truth of a proposition.

Intuitively, an *inform* is used by an agent to assert the truth of the content to another agent. In a commitment-based setting like that of [5], this equates to the speaker agent to *commit* to the truth of the content to the hearer agent.

In our framework, a possible definition of *inform* is as follows:

$$\begin{aligned} &\mathbf{H}(inform(A, B, P, D, T)) \\ \rightarrow &\mathbf{O}(truthful(A, B, P)) \end{aligned} \tag{3}$$

where P is a predicate[4], and with $\mathbf{O}(\mathit{truthful}(A, B, P))$, we mean that the falsity of P would be considered the violation of a commitment of A towards B.

We are aware that verifiability is a problem here: who is supposed to verify the truth of P? According to our approach, built on the principle that it is not acceptable to make assumptions about agents (and, therefore, about their truthfulness) the social infrastructure should be equipped with a knowledge base allowing it to decide the truth value of P. If this is not the case, the only way is probably to associate no obligation to an *inform* act, and let the hearer agent decide about the trustworthiness of the speaker.

Commissive primitives: promise and conditionalPromise. Commissive primitives are used by an agent that wants to commit itself to the execution of some action, possibly specifying a deadline.

A *promise*, like an *inform*, commits the speaker to the truth of the content, but for the former the speaker is responsible for fulfilling it by a physical action.

$$\begin{aligned} &\mathbf{H}(\mathit{promise}(A, B, P, D, T_p)) \\ \rightarrow &\mathbf{O}(\mathit{do}(A, B, P, D, T_d)) : T_d \leq T_p + \tau \end{aligned} \tag{4}$$

The obligation in constraint (4) is *forward*, in that it regards an event that should happen in the future. *do* is a physical action, which fulfills the obligation of deontic constraint (4), provided that it is performed before a certain amount of time τ has passed since the promise, as expressed by the $T_2 \leq T_1 + \tau$ constraint. If no proper *do* action has been performed at the deadline, a violation should be detected.

The obligation in a *conditionalPromise* becomes effective only when an external condition (indicated by Q in the following deontic constraint) becomes true:

$$\begin{aligned} &\mathbf{H}(\mathit{conditionalPromise}(A, B, P|Q, D, T_{cp})) \\ \wedge &\mathbf{H}(\mathit{true}(Q, T_Q)) \\ \rightarrow &\mathbf{O}(\mathit{do}(A, B, P, D, T_d)) : T_d \leq T_Q + \tau \end{aligned} \tag{5}$$

where $\mathit{true}(Q, T_Q)$ represents the fact that, in the instant T_Q, the condition Q becomes true, and this is recognized by the social infrastructure as a significant event.

Directive primitives: request and conditionalRequest. The goal of directive primitives is to invite the hearer agent to perform some action. It is up to the hearer whether to *accept* or to *reject* the content of a directive communicative act.

A *request* does not, by itself, commit the hearer agent to its content, which becomes an obligation only when the request is *accept*ed, as the following deontic

[4] Intuitively, the content should represent a condition that does not depend on the future behavior of agents, as, for instance, $\mathit{sunnyDay}(\mathit{today})$.

constraint states:

$$
\begin{aligned}
&\mathbf{H}(request(B, A, P, D, T_r)) \\
&\wedge \mathbf{H}(accept(A, B, P, D, T_a)) \\
&\wedge T_r < T_a \\
&\rightarrow \mathbf{O}(do(A, B, P, D, T_d)) : T_d \leq T_a + \tau
\end{aligned}
\tag{6}
$$

However, if so desired, it is possible to commit the hearer to answer positively or negatively by a fixed amount of time, by means of following deontic constraint:

$$
\begin{aligned}
&\mathbf{H}(request(A, B, P, D, T_{rq})) \\
&\rightarrow \mathbf{O}(accept(A, B, P, D, T_a) : T_a \leq T_{rq} + \tau) \\
&\vee \mathbf{O}(reject(A, B, P, D, T_{rj}) : T_{rj} \leq T_{rq} + \tau)
\end{aligned}
\tag{7}
$$

The expectations in constraint (7) are *alternative*, i.e., they are fulfilled if at least one of them is.

A *conditionalRequest* is different from a *request* in that its content becomes an obligation only once (the hearer has *accept*ed it and) an external condition has become true:

$$
\begin{aligned}
&\mathbf{H}(conditionalRequest(B, A, P|Q, D, T_{cr})) \\
&\wedge \mathbf{H}(accept(A, B, P|Q, D, T_a)) \\
&\wedge T_{cr} < T_a \\
&\wedge \mathbf{H}(true(Q, T_Q)) \\
&\rightarrow \mathbf{O}(do(A, B, P, D, T_d)) : T_d \leq T_Q + \tau
\end{aligned}
\tag{8}
$$

Proposal primitives: propose. A proposal is a conjunction of a conditional directive and a conditional commissive.

A *propose* is similar to a *conditionalRequest*, with the intuitive difference that for the former the speaker is able to fulfill the condition by a *do* action.

$$
\begin{aligned}
&\mathbf{H}(propose(B, A, (P_A, P_B), D, T_p)) \\
&\wedge \mathbf{H}(accept(A, B, (P_A, P_B), D, T_a)) \\
&\wedge T_p < T_a \\
&\rightarrow \mathbf{O}(do(A, B, P_A, D, T_{d_A})) \wedge \mathbf{O}(do(B, A, P_B, D, T_{d_B})) : \\
&\quad T_{d_A} \leq T_a + \tau_A \wedge T_{d_B} \leq T_a + \tau_B
\end{aligned}
\tag{9}
$$

As for *request* and *conditionalRequest*, however, *propose* does not, per se, generate any obligation (unless it is desired to force an answer as in constraint (7)). It is with *accept* that both the speaker and the hearer become committed to their respective obligations (see constraint (9)).

accept and reject. After receiving a *request*, a *conditionalRequest* or a *propose*, an agent can make the related obligations effective with an *accept* communicative act. This is expressed by constraints (6), (8) and (9).

It should be noticed that obligations are generated only by proper sequences of communicative acts.

It may be desirable to forbid a communicative act unless a certain condition holds. For instance, an *accept* only makes sense in reply to a *request*, *conditionalRequest* or *propose*. For this purpose, *backward* expectations can be used, i.e., expectations about events that *should* (or *should not*) *have happened* when the expectation is generated.

For instance, constraint (10) expresses that, if an *accept* has happened, a corresponding *request*, *conditionalRequest* or *propose* should have happened before. If none of the three holds, a violation should be detected.

$$\begin{aligned} &\mathbf{H}(accept(A,B,P,D,T_a)) \\ \rightarrow &\mathbf{O}(request(B,A,P,D,T_r)) : T_r < T_a \\ \vee &\mathbf{O}(conditionalRequest(B,A,P,D,T_{cr})) : T_{cr} < T_a \\ \vee &\mathbf{O}(propose(B,A,P,D,T_p)) : T_p < T_a \end{aligned} \tag{10}$$

An agent that *reject*s a *request*, *conditionalRequest* or *propose* is obviously not committed to its content. Thus, the only obligation associated with a *reject* is backward, and it is needed to ensure (if so wanted) that a *reject* is only performed in reply to a corresponding *request*, *conditionalRequest* or *propose*.

$$\begin{aligned} &\mathbf{H}(reject(A,B,P,D,T_{rj})) \\ \rightarrow &\mathbf{O}(request(B,A,P,D,T_{rq})) : T_{rq} < T_{rj} \\ \vee &\mathbf{O}(conditionalRequest(B,A,P,D,T_{cr})) : T_{cr} < T_{rj} \\ \vee &\mathbf{O}(propose(B,A,P,D,T_p)) : T_p < T_{rj} \end{aligned} \tag{11}$$

With this semantics of *accept* and *reject*, it is not necessary to impose one more constraint to make the disjunction in constraint (7) exclusive: if an agent both *accept*s and *reject*s a *request*, *conditionalRequest* or *propose*, it is committed to the content anyway. However, if so desired this can be achieved by:

$$\begin{aligned} &\mathbf{H}(accept(A,B,P,D,T_a)) \\ \rightarrow &\mathbf{ON}(reject(A,B,P,D,T_r)) \end{aligned} \tag{12}$$

Since there are no constraints on T_a and T_r, it is not necessary to impose the inverse constraint, because if both an *accept* and a *reject* were issued a violation would be detected, regardless of the order of the communicative acts.

4 Discussion and related work

In defining ACL semantics, we have followed a social approach which, better than the mentalistic, seems to fulfill the requirements of communication in open societies of autonomous and heterogeneous agents.

The syntax of deontic constraints here presented is an extension of that presented in [7], where the model of agent society here assumed is described in more detail, deontic constraints are given a formal declarative semantics and a prototypical implementation of deontic constraints exploiting the *CHR* language is sketched.

The additional syntactical elements here introduced allow us to express *alternative* expectations in the head of a deontic constraint, i.e., expectations which can be fulfilled by different sequences of events, as in constraints (7), (10) and (11).

The implementation of the computational framework described in [7] is currently being expanded in order to manage the syntactical (and semantical) extension.

In part, we draw inspiration from work on commitment-based ACL semantics, notably Fornara and Colombetti's [5], where an operational specification of an ACL is given in an object-oriented framework by means of the *commitment* class.

A commitment is described by a finite state automaton, whose state transitions can occur by application of methods of the commitment class, or of rules triggered by external conditions. Semantics of communicative acts is specified in terms of methods to be applied to a commitment when a communicative act is issued.

The approach there presented is similar to ours in that it is *social-based*: it makes no assumptions on the nature of agents, specifies semantics of actions as their social effects and assumes the existence of a social framework (which is called *institution* in [8]) for assigning agents with roles, verifying agents' social behavior and, possibly, recovering from violation conditions.

There are, however, some significant differences, originating, in part, from the different paradigms of the two frameworks (object-oriented and logic-based). For instance, a commitment can have several states, while expectations do not have states. Although some of the possible states of a commitment do not have a direct mapping to our framework, which could make our framework seem less expressive, as explained in [7] we can express an equivalent semantics (in terms of fulfillments and violations with respect to a sequence of communicative and physical acts) to one expressed in the framework of [5]. In general, it is sufficient to specify the sequences of communicative actions that should raise an expectation.

Furthermore, social semantics of communicative acts in Fornara and Colombetti's framework has a unique definition as a method application on a commitment: for instance, an *accept* will always turn a *pre-commitment* into a *conditional commitment*. In our framework, instead, it is possible to specify that a communicative act generates an expectation only in a given combination of events, thus allowing to give "context-sensitive" semantics to communicative acts.

Our notion of *expectation* is also more general than that of *commitment*: it represents the deontic necessity of a (past or future) event, and is not bound to have a debtor or a creditor, or to be brought about by an agent. For instance, if we want to express the constraint that an agent is only *allowed* to perform an act when a previous event has occurred (not necessarily for an agent's action), we can simply impose it as a backward obligation (as, for instance, constraint (10)), whereas it is not obvious how to express this in Fornara and Colombetti's framework [5]. In other words, the formalism of deontic constraint can express both social semantics of communicative acts and interaction protocols. In [9], for instance, we exploit deontic constraints for expressing protocols for an agent society, rather than the semantics of communication.

The deontic framework can, in part, be integrated with dialogue theory [10], where different types of dialogues which can occur between agents are modeled in terms of *initial situation*, *private aims* of the interlocutors and *joint aims* implicitly subscribed by the interlocutors. Deontic constraints can express the allowed sequences of utterances to compose a dialogue framework [11]. Moreover, the result of dialogues can raise expectations about the subsequent behavior of agents: for instance, once a collective intention [12] is established, the agents can be expected to fulfill it. However, since we make no assumptions about the mental structure of individual agents, the results of dialogues regarding beliefs (namely *persuasion*, *inquiry* and *information seeking*) are not verifiable in our framework.

Future work will regard both the declarative specification of a society of agents and its operational counterpart.

Declaratively, current work is devoted to the problem of deciding which agent is responsible for a violation of an expectation, which is not obvious in the general case. This is related to the definition of the recovery procedures mentioned in Sect. 2, which is, too, subject of current work.

Operationally, we plan to extend the prototypical implementation of deontic constraints [7] in order to handle alternative expectations, and to improve and refine it as needed to operate in a real world open environment.

In addition, we intend to investigate more deeply the formal properties of the proof procedures involved in the framework, such as soundness and completeness with respect to the declarative specification as done, for instance, in [13].

We are also considering the possibility of integrating deontic constraints as integrity constraint in a more general abductive framework.

Acknowledgments

This work is partially funded by the Information Society Technologies programme of the European Commission under the IST-2001-32530 project (SOCS). The information provided is the sole responsibility of the authors and does not reflect the Commission's opinion. The Commission is not responsible for any use that might be made of the data appearing in this publication.

References

1. Finin, T., Labrou, Y., Mayfield, J.: KQML as an agent communication language. In: Software Agents. MIT Press, Cambridge (1997)

2. FIPA: FIPA Communicative Act Library Specification (2001) Published on August 10th, 2001, available for download from the FIPA website: `http://www.fipa.org`.
3. Singh, M.: Agent communication language: rethinking the principles. IEEE Computer (1998) 40–47
4. Singh, M.P.: A social semantics for agent communication languages. In Dignum, F., Greaves, M., eds.: Issues in Agent Communication. Springer-Verlag, Heidelberg, Germany (2000) 31–45
5. Fornara, N., Colombetti, M.: Operational specification of a commitment-based agent communication language. In Castelfranchi, C., Lewis Johnson, W., eds.: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2002). Part II, Bologna, Italy, ACM (2002) 535–542
6. Yolum, P., Singh, M.: Commitment machines. In: Intelligent Agents VIII, 8th International Workshop, ATAL 2001 Seattle, WA, USA, August 1-3, 2001 Revised Papers. Volume 2333 of LNAI., Springer Verlag (2002) 235–247
7. Alberti, M., Ciampolini, A., Gavanelli, M., Lamma, E., Mello, P., Torroni, P.: Logic based semantics for agent communication languages. In Dunin-Keplicz, B., Werbrugge, R., eds.: Proceedings of the Workshop of Formal Approaches to Multiagent Systems (FAMAS03), Warsaw, Poland. (2003)
8. Colombetti, M., Fornara, N., Verdicchio, M.: The role of institutions in multiagent systems. In: Proceedings of the Workshop on Knowledge based and reasoning agents, VIII Convegno AI*IA 2002, Siena, Italy. (2002)
9. Torroni, P., Mello, P., Maudet, N., Alberti, M., Ciampolini, A., Lamma, E., Sadri, F., Toni, F.: A logic-based approach to modeling interaction among computees (preliminary report). In: UK Multi-Agent Systems (UKMAS) Annual Conference, Liverpool, UK. (2002)
10. Walton, D., Krabbe, E.: Commitment in Dialogue. SUNY, New York (1995)
11. Reed, C.A.: Dialogue frames in agent communication. In: Proceedings of the 3rd International Conference on Multi Agent Systems (ICMAS98), Paris, IEEE Press (1998) 246–253
12. Dignum, F., Dunin-Keplicz, B., Verbrugge, R.: Creating collective intention through dialogue. Logic J. of the IGPL **9** (2001) 145–158
13. Ciampolini, A., Lamma, E., Mello, P., Torroni, P.: An operational semantic for the safe execution of tasks in a constrained multi-agent setting. In Greco, S., Leone, N., eds.: Proceedings of the 8th European Conference on Logics in Artificial Intelligence (JELIA). Volume 2424 of LNCS., Springer Verlag (2002) 14–26