A Social Approach to Communication in Multiagent Systems *

Marco Colombetti[1,2], Nicoletta Fornara[1], and Mario Verdicchio[2]

[1] Università della Svizzera italiana, via Buffi 13, 6900 Lugano, Switzerland
{nicoletta.fornara, marco.colombetti}@lu.unisi.ch
[2] Politecnico di Milano Piazza Leonardo Da Vinci 32, 20133 Milano, Italy
{mario.verdicchio, marco.colombetti}@elet.polimi.it

Abstract. This paper aims at defining the semantics of Agent Communication Languages (ACLs) in terms of changes in the social relationships between agents, represented in terms of *social commitments*. We take commitment to be a primitive concept underlying the social dimension of multiagent systems, and define a basic artificial institution that provides agents with the means to affect the commitment network that binds them to each other. Two different approaches are adopted for the presentation of our proposal: a logical formalization and an operational specification.

1 Introduction

Since the beginning of the 1990s, the community of Multiagent Systems (MAS) researchers has carried out significant efforts to design a standard, application-independent language for agent communication. As is well known, different approaches to agent communication have been advocated. While all competing proposals share the view that an Agent Communication Language (ACL) should be based on the concept of a *communicative act*, at least four different views of ACL semantics have been put forward and defended. Such views can be briefly described as follows:

- the *mentalistic* view: the semantics of a communicative act is defined in terms of its effects on the mental states of the communicating agents [1], [2], [3];
- the *social* view: the semantics of a communicative act is defined in terms of its effects on the social states binding the communicating agents [4], [5], [6];
- the *protocol-based* view: the semantics of a communicative act is defined in terms of the conversational protocols in which it occurs [7];
- the *conventional* view: the semantics of a communicative act is defined in terms of a set of conventions, which operate under a principle of informational optimality [8];

* Partially supported by Swiss National Science Foundation project 200021-100260, "An Open Interaction Framework for Communicative Agents"

Elsewhere we have advocated a social approach to agent communication, and proposed both a logical [9] and an operational model [10] [11] of ACL semantics. In this paper we try to bring the two models together. We start from the assumption that communication is, above all, an *institutional activity*, that is, a kind of social activity regulated by a collection of institutions. By an *institution* we mean a set of concepts and rules shared and jointly accepted by a group of agents. As we shall see, institutions provide a fundamental component of the context in which communication is carried out. If we regard communicative acts as institutional actions [12] (i.e., actions that are made possible by the existence of a number of institutions), an ACL is just a set of conventions to carry out institutional actions. The treatment of language as a conventional means to perform institutional actions is of course inspired by Speech Act Theory, in particular as presented by Searle [13]. However, in view of our specific goal we do not try to base our treatment on a model of human institutions; rather, we take them as a source of inspiration to define a concept of institution that suits artificial agents.

As we said before, this paper presents a model of agent communication first in logical and then in operational terms. This choice deserves some justification. There are indeed different reasons to define a model of agent communication. A first reason is to specify the semantics of an ACL so that the act of sending a message of a given form has unambiguous preconditions and effects. A further reason is to establish a basis for proving properties of ACL-related constructions, like for example conversation protocols. Yet another reason is to provide a clear guidance to the implementation of multiagent platforms allowing agents to interact through a standard ACL. Declarative models, like the one built on logic, are best suited to give a rigorous definition of ACL semantics, and can also be very useful to prove properties of conversation protocols, for example by model-checking techniques. On the other hand, declarative models do not provide useful guidance to implementation: to this purpose, an operational specification is more suitable.

In principle, the most natural approach to this kind of multiple modeling would be to develop a logical model first, and then to use it as the basis of an operational specification. However, this is not how our research activity has developed in the last few years. In fact, for practical reasons we have developed both models in parallel; as a result the two models, as they have been presented so far in the literature ([10], [11], [9]) are not completely consistent. In this paper we try to adapt our models to achieve higher consistency; however, it should be noted that a logical model and an operational specification are not just two different views of the same thing. To clarify this point, let us consider as an example a logical and an operational specification of arithmetic. Natural numbers are logically modeled by Peano axioms; however, such axioms say nothing on the computer representation of a natural number, on how to perform arithmetical operations in a digital device, and so on. Digital number crunching is often just an approximation to Peano arithmetic, for example when it implements modular arithmetic on a finite set of numbers, due to memory limitations. Analogously, the logical model and the operational specification presented in this paper are meant to highlight aspects of agent communication that are at least in part different.

To be more specific, let us now see where the two models we propose in this paper depart from each other. In the logical model we define the preconditions and effects of communicative acts in terms of *truth conditions*; in other words, we specify the state of affairs that holds in a specific kind of model structure before and after a communicative act is successfully performed. In principle, such a declarative specification could be directly implemented in an artificial agent, provided the agent has full theorem proving capacities with respect to a first order temporal logic with branching time: an assumption that would bring us very far from the current agent technology. On the contrary, when we define our operational specification we consider a set of data structures and operations that actual agents may easily implement, without relying on theorem proving techniques. In other words, while a logical model is fit for reasoning on agent systems, an operational specification provides guidance on how to realize actual agents; nevertheless, the relationship between the logical model and its operational specification is still very tight, the latter being a realistic approximation to the former. It must be noted that a different approach may be followed. Indeed, as suggested by [14], a declarative representation may be effectively used by actual agents: this approach may provide a viable alternative to operational specifications, and allow for a more direct use of at least part of a logical model. However, to understand which of the two approaches is more productive requires further research.

The rest of this paper is structured as follows. In Section 2 we introduce the fundamental concepts underlying our treatment of agent communication. In Section 3 we present our logical model of agent communication. We then show in Section 4 how the model can be specified operationally. Finally, in Section 5 we draw some conclusions.

2 The Conceptual Framework

Communicative acts are events that conventionally take place when messages are exchanged in a given context. As all events, communicative acts have effects, some of which correspond to the perlocutionary effects of Speech Act Theory [13]; for example, by informing agent b that an auction is going to close in ten minutes, agent a may significantly affect the behavior of agent b. Perlocutionary effects are often the *reason* for performing a communicative act, but they do not *define* communicative acts; analogously, one may close a door to keep a room warm, but keeping a room warm does not define an action of closing a door.

Being conventional, communicative acts can only be defined in terms of institutional effects; more specifically, we regard communicative acts as institutional actions performed by way of exchanging messages. This fact is obvious for acts of declaration, like for example declaring war or declaring a meeting open: in such cases the effects (war is declared, a meeting is open) are clearly institutional.

Commissives, like for example promises, are another example of communicative acts whose effects are easily regarded as institutional: intuitively, a commissive act brings about some kind of obligation, and obligations are part of institutional reality. Less clear is how we may regard as institutional the effects of assertives, like acts of informing, or directives, like requests. This point will be developed in due course.

Regarding communicative acts as institutional presupposes a clear definition of the concept of an *institution*. By this term we mean a set of shared concepts and rules that regulate the management of a fragment of social reality [15],[16]. Suppose for example that agent a sends an order to agent b. For the order to be felicitous, it is necessary that the two agents are part of a hierarchical organization, empowering a to issue an order to b; no such organization is presupposed, on the contrary, by an act of informing or requesting. In human society, an institution can be based on spontaneous agreement (like, for example, in the case of a promise binding two individuals) or on an explicit body of regulations (like, for example, in the case of marriage in modern societies). In the case of multiagent systems, we can assume that all institutions will have to be explicitly and formally represented. The general picture we have in mind is the following. To play a part in multiagent systems, an agent must be officially recognized as a member of the “society of agents”. To achieve this, the agent will first have to undergo a suitable registration procedure; if the procedure is carried out successfully, the agent will be allowed to interact with other registered agents according to the *Basic Institution*, that is, the institution setting the general concepts and rules of agent interaction.

As we shall see in the rest of the paper, the Basic Institution provides enough rules to perform a set of fundamental communicative acts, like informing, requesting, promising, and so on. Many kinds of interaction, however, require a richer social structure; as we have already remarked, for example, orders presuppose a hierarchical organization, which is not part of the Basic Institution. To deal with such cases we assume that further institutions, which we call *special institutions*, can be invoked; to make an example, electronic commerce will only be possible on the ground of special institutions regulating ownership and money.

According to our standpoint, the fundamental function of the Basic Institution is to regulate the management of *social commitments* between agents. By doing so, the Basic Institution provides for the ontological ground that is necessary and sufficient to define all communicative acts, with the possible exception of those relying on special institutions. We view commitments as institutional states that bind two agents (the creditor and the debtor of the commitment) and can be made, accepted, refused or cancelled by agents, provided they have sufficient institutional powers to do so. Moreover, a commitment may be pending, fulfilled or violated according to the fact that its content is undefined, true or false in the world.

Once the ontology of commitment is established, and the relevant institutional powers of agents are defined, it is possible to specify the effects of communicative acts in terms of commitments; in turn, the effects of communicative acts can be used to define the semantics of a library of ACL messages. By itself, however, message semantics is not sufficient to guarantee effective communication. In an electronic auction, for example, agents exchange meaningful messages according to a rigid protocol, which is itself part of the definition of an auction. Coherently with our standpoint, we view an auction as an example of a special institution that, among other things, specifies a set of *norms* to regulate the interactions of the participants. Such norms are typically, even if not necessarily, specified as an interaction protocol.

In the next section we define a logical model of the concepts we have just described in informal terms.

3 The Logical Model

We base our logical model on a first-order temporal language (which we call the *Semantic Language*, SL), that can be regarded as a metalanguage for the definition of an ACL. Coherently with all major proposals in this area, we deal with agent communication in terms of communicative acts, which we define as changes at the level of social commitments between agents. We take *commitment* to be a primitive concept that underlies the social structure of a multiagent system, and describe communicative acts as actions performed by an agent to affect the commitments that bind it to other agents. Agents communicate by message exchanges, which *count as* communicative acts only when some particular conditions hold within the context of an *artificial institution*; in particular, we call *Basic Institution* the institution that sets the general concepts and rules of agent interaction.

We articulate the presentation of our model in a number of sections. More precisely, in Section 3.1 we give a brief description of the formalism we use to describe communicative acts. Sections 3.2 to 3.6 describe the Basic Institution, which enables agents to act as members of a common society. Section 3.7 gives a general view on how all the components of the Basic Institution affect agent communication by representing the performance of communicative acts in a general form. Finally, Section 3.8 illustrates how it is possible to execute any communicative act in the form of a declaration.

3.1 Time and action

Actions are events brought about by agents. When agent a brings about an event (or performs an action) e of type τ, the SL formula

$$Done(e,a,\tau)$$

holds. We sometimes use the 'n-dash' character to express existential quantification, as in the examples below:

$$Done(e,-,\tau) =_{def} \exists x\, Done(e,x,\tau);$$
$$Done(e,-,-) =_{def} \exists x\, \exists \tau\, Done(e,x,\tau).$$

To deal with events, SL includes a temporal logic with CTL*-like operators [17]. We assume that the logical model relies on a discrete-time frame with a tree-like structure, infinite both in the future and in the past, in which every state has a unique predecessor and at least one successor, and there is at most one walk between any pair of states. A *path* is an infinite sequence $p = \langle p_0,...,p_n,...\rangle$ of states, such that for every element p_i in the sequence, element p_{i+1} is one of the successors of p_i in the frame. All formulae on path p are evaluated with respect to p_0, which is the *starting point* of the path. Paths allow us to formalize the concepts of being "in the past" or "in the future" of some state. More precisely, we say that state s' is in the future of s if and only if there is a path p such that $s = p_0$ and, for some n, $s' = p_n$. Symmetrically, we say that s' is in the past of s if and only if there is a path p such that $s' = p_0$ and, for some n, $s = p_n$. We assume that on every path an event can happen only once, as stated by the following axiom:

(UE) $Done(e,\text{-},\text{-}) \rightarrow \mathsf{G}^-\mathsf{X}^-\neg Done(e,\text{-},\text{-}) \land \mathsf{AG}^+\mathsf{X}^+\neg Done(e,\text{-},\text{-})$.

Here we use some of the following temporal operators, whose intuitive meanings are illustrated below:

A for all paths;
G^+ always on the current path, at the current state and in the future;
G^- on the current path at the current state and in every state that is in the past of the path's starting point;
X^+ at the next state on the current path;
X^- at the previous state on the current path;
F^+ at the current state on the current path, or sometime in the future;
F^- at the current state on the current path, or in some state in the past of the path's starting point.

A complete account of the temporal logic in which we embed our communication framework can be found in [9].

Communicative acts are actions that agents bring about to communicate with other agents. We regard them as institutional actions performed by way of exchanging messages. More precisely, we rely on the concept of *commitment* ([4],[5],[6]) to specify the conventional effects of message exchanges. Commitments are part of the social reality defined and regulated by the Basic Institution. Some work in the direction of defining institutions in general has already been carried out in the field of multiagent systems (see for example [18], [19]). For our purposes, we define an institution as comprised of four fundamental components:

- *core ontology*: the ontology of the social concepts defined by the institution;
- *authorizations*: the specification of the institutional effects that each member of the institution is empowered to bring about, typically authorizations are associated with *roles* in order to abstract from specific agents;
- *norms*: the obligations and the permissions imposed by the institution to its members;

– *conventions*: the specification of the relation between concrete events and their institutional effects.

The Basic Institution regulates the general aspects of communicative interaction, including commitments, as we explain in the sequel.

3.2 Commitment

In our approach, the core ontology of the Basic Institution defines *commitment*, *precommitment*, and a set of operations for *commitment manipulation*.

Commitment and precommitment Commitment is a primitive concept underlying the relations between agents. More precisely, a commitment holds in a state in which an agent (the *debtor*) is bound, relative to another agent (the *creditor*), to the fact that some proposition (the *content*) is true. The content of a commitment is a sentence of a *Content Language* (CL), represented as a first-order term of SL. The relevant formula is

$Comm(e,x,y,s)$,

which states that communicative act e has brought about a situation that binds agent x, relative to agent y, to the truth of a proposition of the content language, represented by the SL term s. When a commitment is proposed to an agent (normally, the potential debtor) for acceptance, but it has not been accepted nor refused yet, we say that a *precommitment* holds. Precommitments are represented analogously to commitments:

$Prec(e,x,y,s)$

holds when e has brought about a precommitment between two agents (the potential debtor, x, and the potential creditor, y) to the truth of a CL sentence represented by s. We regard the creation and the modification of (pre)commitments as the effects of communicative events that have to be dealt with to define an effective ACL semantics.

As we already said in Section 2, a commitment may be pending, fulfilled or violated according to the fact that its content, s, is undefined, true or false in the world. Defining the truth conditions for sentences in our model is not trivial, mainly because of the branching structure of time. This issue is explored in more detail in Section 3.3.

Commitment manipulation Commitments and precommitments arise from the performance of communicative acts. More precisely, agents bring about communicative events by exchanging messages; a communicative event, under given conditions, counts as a *commitment manipulation action*, which creates a new (pre)commitment or modifies an existing one.

The core ontology of the Basic Institution allows for five basic operations for commitment manipulation. More specifically, agents can: make a commitment (by performing action $mc(x,y,s)$), make a precommitment ($mp(x,y,s)$), cancel a commitment ($cc(e,x,y,s)$), cancel a precommitment ($cp(e,x,y,s)$), or accept a precommitment, that is, turn it into a commitment ($ap(e,x,y,s)$). Such action types are defined by axioms that describe their constitutive effects, that is, the state of affairs that necessarily hold after a token of the given action type is successfully performed.

We now introduce further temporal operators to express our axioms in a simpler form:

$\varphi\ \mathsf{U}^+\psi$ (φ is true until ψ is eventually true);

$\varphi\ \mathsf{W}^+\psi =_{def} \mathsf{G}^+\varphi \vee \varphi\ \mathsf{U}^+\psi$ (weak until operator);

$\varphi\ \mathsf{Z}^+\psi =_{def} \varphi\ \mathsf{W}^+\psi \wedge \mathsf{G}^+(\psi \rightarrow \mathsf{G}^+\neg\varphi)$ ($\varphi\ \mathsf{Z}^+\psi$ is true if and only if in the future ψ never becomes true and φ is always true, or φ is true until ψ eventually becomes true and since then φ is no longer true).

We are now ready to present the axioms that describe the commitment manipulation actions.

(MC) $Done(e,\text{-},mc(x,y,s)) \rightarrow \mathsf{A}(Comm(e,x,y,s)\mathsf{Z}^+\ Done(\text{-},\text{-},cc(e,x,y,s)))$.

Axiom MC (Make Commitment) states that:

> if an agent (not necessarily x or y) performs an action of making a commitment with x as the debtor, y as the creditor, and s as the content,
> then, on all paths, x is committed, relative to y, to the truth of s,
> until an agent possibly cancels such a commitment, after which the commitment no longer exists.

(MP) $Done(e,\text{-},mp(x,y,s)) \rightarrow$
$\mathsf{A}(Prec(e,x,y,s)\ \mathsf{Z}^+(Done(\text{-},\text{-},ap(e,x,y,s)) \vee\ Done(\text{-},\text{-},cp(e,x,y,s))))$.

Axiom MP (Make Precommitment) is analogous to MC.

(AP) $Done(e',\text{-},ap(e,x,y,s)) \rightarrow \mathsf{A}(Comm(e',x,y,s)\mathsf{Z}^+ Done(\text{-},\text{-},cc(e',x,y,s)))$.

Axiom AP (Accept Precommitment) implies that if an agent performs an action of accepting a precommitment brought about by event e with x, y, and s respectively as debtor, creditor, and content, then the acceptance action brings about, on all paths, a commitment of x, relative to y, to the truth of s, which will stand until it is possibly cancelled.

There are no specific axioms for the actions of canceling a precommitment (cp) or a commitment (cc), because the analytical effects of these commitment manipulations are already illustrated in the axioms dealing with other actions, whose performance must be presupposed in order to take into account any kind of cancellation. This issue is dealt with in more detail in the next section.

Ontological preconditions Some of the commitment manipulation actions rely on *ontological preconditions*, that is, they can be performed only if particular states of affairs hold. For instance, the ontological precondition to a commitment's cancellation is the existence of the commitment itself. The formula

$OntPoss(\tau)$

states that the ontological preconditions of an event of type τ hold; we also say that the event is *ontologically possible*. We assume that making a commitment or a precommitment is always ontologically possible, as specified by the following axioms:

(PMC) $OntPoss(mc(x,y,s))$,
(PMP) $OntPoss(mp(x,y,s))$.

The axioms dealing with the preconditions of the actions of cancelling and accepting are the following:

(PCC) $OntPoss(cc(e,x,y,s)) \leftrightarrow \mathsf{X}^- Comm(e,x,y,s)$;
(PCP) $OntPoss(cp(e,x,y,s)) \leftrightarrow \mathsf{X}^- Prec(e,x,y,s)$;
(PAP) $OntPoss(ap(e,x,y,s)) \leftrightarrow \mathsf{X}^- Prec(e,x,y,s) \wedge \neg Done(\text{-},\text{-},cp(e,x,y,s))$.

The core ontology also describes the states (*fulfilled*, *violated*, *pending*) in which a commitment can be. The definition of these states relies on the *truth conditions* of the content of the commitment, which are formally described in the next subsection.

3.3 The representation of content

Before we define the truth conditions of CL sentences, two remarks should be made. First, the truth of a temporal sentence at a given state (the *point of reference*, [20]) can be evaluated only if we know at which state the sentence has been uttered (the *point of speech*). For example, the sentence "I shall pay you within the end of the month" implicitly refers to the end of the current month, which in turn is determined by the state at which the sentence is uttered. Second, branching time brings in a semantic difficulty known as *contingent future*, which means that at a given point of reference it may be still undetermined if a sentence is going to be true or false. Consider again to the previous example, and assume that the sentence has been uttered, on January 10^{th}, and that no payment has been made as far as January 15^{th}; on January 15^{th}, the sentence is still not settled true nor settled false, and thus it is undefined. Note however that a sentence that is true (false) at a state, will go on being true (false) at all states in the future of that state.

We represent CL sentences as SL terms, which allows us to define CL semantics in SL. More precisely, we first assume that for every SL term s that denotes a CL sentence, there is exactly one SL formula $\lfloor s \rfloor$ that corresponds to s, which we call the *sentence meaning* of s; then we define the truth conditions of s in SL.

CL semantics is dealt with by means of the following predicates, whose definitions we call *truth conditions* of a sentence:

$True(e,s) \leftrightarrow \mathsf{AF}^-(Done(e,\text{-},\text{-}) \wedge \lfloor s \rfloor)$,
$False(e,s) \leftrightarrow \mathsf{AF}^-(Done(e,\text{-},\text{-}) \wedge \neg \lfloor s \rfloor)$,
$Undef(e,s) \leftrightarrow \mathsf{AF}^- Done(e,\text{-},\text{-}) \wedge \neg True(e,s) \wedge \neg False(e,s)$.

Note that the A path quantifier in front of the F^- operator is necessary, even if our model structure is not branching in the past, because formula $\lfloor s \rfloor$ may include operators like F^+ that need a path to be specified.

The truth conditions of sentence s are given with respect to an event e, which does not necessarily correspond to the event of uttering s. Event e is used to set a well-defined temporal reference by which we can evaluate the truth of s. The truth conditions of a CL sentence determine the fulfillment or the violation of the relevant commitment. More precisely, a commitment whose content is s is said to be *fulfilled*, *violated*, or *pending* respectively when s is true, false, or undefined according to the above definitions. The event with respect to which the truth conditions of the content are checked is the one that has brought about the commitment. Here are the axioms that formalize what stated above:

$$\begin{aligned}&\mathit{Fulf}(e,x,y,s) \leftrightarrow \mathit{Comm}(e,x,y,s) \wedge \mathit{True}(e,s),\\&\mathit{Viol}(e,x,y,s) \leftrightarrow \mathit{Comm}(e,x,y,s) \wedge \mathit{False}(e,s),\\&\mathit{Pend}(e,x,y,s) \leftrightarrow \mathit{Comm}(e,x,y,s) \wedge \mathit{Undef}(e,s).\end{aligned}$$

Intuitively, every commitment is either fulfilled, or violated, or pending. It is actually possible to prove that

$$\models \mathit{Comm}(e,x,y,s) \rightarrow \mathsf{xor}(\mathit{Fulf}(e,x,y,s), \mathit{Viol}(e,x,y,s), \mathit{Pend}(e,x,y,s)).$$

This means that only one of $\mathit{Fulf}(e,x,y,s)$, $\mathit{Viol}(e,x,y,s)$, or $\mathit{Pend}(e,x,y,s)$ is true in all models in every state in which $\mathit{Comm}(e,x,y,s)$ holds.

The core ontology delimits the set of possible institutional actions. However, not every ontologically possible action can be actually carried out: it is part of the function of an institution to constrain the execution of the actions that are ontologically possible.

3.4 Authorizations

In order to interact with other agents within the context of the Basic Institution, an agent must be *registered*. We assume that every registered agent plays a specific role, *RegAgt*, within the Basic Institution (*BI*). If a is a registered agent, the SL formula

$$\mathit{Role}(a,BI,\mathit{RegAgt})$$

holds.

We now introduce a predicate to state that an agent is authorized to bring about an event of a certain type. In general, to effectively perform an institutional action of type τ, agent x must be authorized to do so, that is, the formula

$$\mathit{Auth}(x,\tau)$$

must hold. A reasonable set of authorizations concerning the creation and the manipulation of commitments can be defined in the form of an axiom as follows:

(ABI) $Role(x{,}BI{,}RegAgt){\wedge}Role(y{,}BI{,}RegAgt){\rightarrow}$
$Auth(x{,}mc(x{,}y{,}s)){\wedge}\ Auth(x{,}mp(y{,}x{,}s)){\wedge}\ Auth(x{,}cp(e{,}x{,}y{,}s)){\wedge}$
$Auth(x{,}cp(e{,}y{,}x{,}s)){\wedge}\ Auth(x{,}cc(e{,}y{,}x{,}s)){\wedge}\ Auth(x{,}ap(e{,}x{,}y{,}s)).$

This formula means that: as a *debtor*, a registered agent is authorized to make a commitment with any registered agent as the creditor, and accept or cancel an existing precommitment; as a *creditor*, a registered agent has the authorization to make precommitments with any registered agent as the debtor, and cancel an existing commitment or precommitment.

All the communicative acts that comply with the authorizations defined above are institutional actions, whose execution is authorized by the Basic Institution. Other types of communicative acts, however, may require a special institution. Consider for example the act of ordering: we view it as a request that cannot be refused, that is, we can model it as creating a commitments as the creditor. If we want to follow the example of human society, orders will not be authorized by the Basic institution, and will only be effective if some special institutional framework, involving a hierarchy, is defined.

3.5 Norms

Authorizations define the institutional powers of agents; in general, however, the exercise of such powers is further regulated by a set of norms. Consider, for example, a scientific society: the president of the society typically has the power to call the general meeting of the society's members; the president, however, is also *obliged* to call such a meeting at least once a year, and may be *allowed* to call it more often if he or she has good reasons to do so. At the present stage of our research, we still do not whether some norms should be regarded as part of the Basic Institution. Consider for example the adjacency pair made up by a request and its acceptance or refusal. At some institutional level, we might want to dictate that agents should react to all requests by producing an acceptance or a refusal. But is this rule to be regarded as a norm of the Basic Institution? Or does it belong to a special institution, like for example an "Institution of Conversations"? We feel that more work has to be done in order to clarify this issue; in particular, practical applications will have to be analyzed to get a better understanding of the systems of norms involved in the functioning of real multiagent systems. For the time being, therefore, we assume that the Basic Institution specifies no norms.

3.6 Conventions

An institutional action is performed through the execution of some lower level act that conventionally counts as a performance of the institutional action; obvious examples are offered by communicative acts, which are performed by executing lower level acts of message exchange. As a consequence, institutional actions require a set of conventions for their execution. The institutional actions of the

Basic Institution are commitment-manipulations actions, conventionally realized by the exchange of ACL messages.

In Section 3.4 we have illustrated the six authorizations that registered agents are granted by the Basic Institution. Now we deal with the structure of the messages that agents actually exchange to perform authorized commitment manipulation actions. We view a message as a pair made up by a *type indicator* and a *body*. Type indicators (analogous to KQML's performatives [2]) are constant symbols taken from a finite set. The body of a message is usually a CL sentence represented in our semantic language by a first-order term. In the case of acceptance or refusal messages, the body is comprised of a more complex structure, that is, a tuple of elements ($\langle e,x,y,s \rangle$) that identifies an existing (pre)commitment. For every message type we introduce a functor that specifies the relevant type of the action that an agent performs when exchanging a message of such a type. This approach is best explained by an example. Suppose that agent x sends a message to agent y to inform y that σ is the case (where σ is a suitable first-order formula). The exchange of such a message is an event of type $inform(x,y,s)$, where *inform* is a three-place functor denoting the type of the message, x and y respectively denote the sender and the receiver of the message, and s is a term corresponding to formula σ. When event e is an exchange of a message of type *inform* and content s, sent by agent x to agent y, the formula

$$Done(e,x,inform(x,y,s))$$

holds. This event, under given conditions, implies the performance of a commitment manipulation action. In other words, the meaning of the message is defined as the effect that exchanging such a message has on the network of commitments binding the sender and the receiver. The correspondence between the type τ of the message exchange and the type τ' of the commitment manipulation action is defined by a convention of the relevant institution, and is formally stated by means of the formula

$$CountAs(\tau,\tau'),$$

which means that an action of type τ conventionally counts as an action of type τ'. Below we define the communicative acts by means of which agents carry out the commitment manipulation actions authorized by the Basic Institution.

Informing is defined as committing to the truth of the message body, which, we suppose, is comprised of an arbitrary CL sentence. More precisely, when agent x exchanges with agent y an *inform* message with content s, agent x commits, relative to y, to the truth of s:

(CAInf) $CountAs(inform(x,y,s),mc(x,y,s))$.

We assume that the body of a *request* message is comprised of an *action expression*, which indicates the requested action's type, its actor, and possibly a temporal constraint. More precisely, in the request message we use a term that represents the abstract syntax of an action expression, which is not to be confused with its concrete form, which belongs to a specific CL. In our SL, an action expression may have the form $Done(x,\tau)\mathsf{B}^+\Gamma$, in which B^+ means intuitively "before" ($\varphi\mathsf{B}^+\psi =_{def} \neg(\neg\varphi\ \mathsf{U}^+\ \psi)$), and Γ is an SL formula referring to a particular time-point. Here we are assuming that for every time-point expression of a CL there exists an SL formula which becomes periodically true according to the time-point it is indicating. Thus, in this case the body of a request message is comprised of the term $\lceil Done(x,\tau)\mathsf{B}^+\Gamma\rceil$, in which $\lceil\rceil$ is a function which, given an SL formula φ, returns the SL term $\lceil\varphi\rceil$ such that $\lfloor\ \lceil\varphi\rceil\ \rfloor$ is φ. If we denote such term with s, and then define the type *request*(x,y,s) to denote events by which agent x requests s from agent y, the semantics of *request* messages is defined as below:

(CAReq) $CountAs(\mathit{request}(x,y,s), \mathit{mp}(y,x,s))$.

The above-mentioned action expression is only an example. We think that the task of defining other action expressions is to be tackled only when we deal with the application of our framework to actual cases.

The act of *accepting* is not only defined with respect to requests, but with respect to precommitments in general. We assume that the body of an acceptance message is a tuple which includes all the elements that uniquely identify the relevant precommitment. We introduce the functor *accept*(e,x,y,s), whose arguments are the same as those characterizing the precommitment that the sender of the message is accepting, and the relevant convention is as follows:

(CAAcc) $CountAs(\mathit{accept}(e,x,y,s), \mathit{ap}(e,x,y,s))$.

The body of a *cancel* message is supposed to be the same as that of an *accept* message, that is, a tuple which includes all the elements that uniquely identify the relevant (pre)commitment. To denote the event types corresponding to such message exchange we introduce the functor *cancel*(e,x,y,s), whose arguments are the same as those characterizing the (pre)commitment which is cancelled by the relevant message exchange. The conventions are then defined like this:

(CACancP) $CountAs(\mathit{cancel}(e,x,y,s), \mathit{cp}(e,x,y,s))$,
(CACancC) $CountAs(\mathit{cancel}(e,x,y,s), \mathit{cc}(e,x,y,s))$.

A *cancel* message exchange can count as different commitment manipulation actions (*cp* and *cc*), in accordance to what is being cancelled (a precommitment or a commitment, respectively). There is no ambiguity in an actual *cancel* message exchange, as there cannot exist both a precommitment and a commitment with the same arguments, and only an action that cancels an existing object can be successfully carried out, as stated by Axioms PCC and PCP in Section 3.2. These ontological preconditions contribute to the general definition of institutional actions, as illustrated in the next subsection.

3.7 The general representation of communicative acts

We are now ready to give a general definition of communicative acts. The *conventions* of the Basic Institution establish that exchanging a message of given type *counts as* a specific institutional action, provided certain conditions hold. These conditions can be classified in two categories: *ontological preconditions*, defined by the *core ontology*, and *authorizations*. While authorizations deal with the institutionalized power of agents, ontological preconditions concern the state of affairs that must hold for a communicative act to be possible. All these points are formally expressed by the following axiom that gives a general definition of institutional actions:

(IA) $Done(e,x,\tau) \wedge CountAs(\tau,\tau') \wedge OntPoss(\tau') \wedge Auth(x,\tau') \rightarrow Done(e,x,\tau')$.

As an example, let us consider the act of informing. Suppose that the formula

$Done(e_1,a,inform(a,b,s_1))$

holds, that is, agent a informs b that s_1 is the case. From Axiom CAInf we derive $CountAs(inform(a,b,s_1),mc(a,b,s_1))$, and thus we determine the commitment manipulation action that corresponds to such a message exchange. We suppose that formulae

$Role(a,BI,RegAgt)$ and
$Role(b,BI,RegAgt)$

hold, that is, both a and b are registered agents in the Basic Institution. Given these premises and Axiom ABI, which illustrates what actions registered agents are authorized to perform, we have (among other consequences)

$Auth(a,mc(a,b,s_1))$.

Axiom PMC states that it is always ontologically possible to perform a make commitment action, thus, from the premises above and Axiom IA we can derive

$Done(e_1,a,mc(a,b,s_1))$,

which, thanks to Axiom MC, gives

$Comm(e_1,a,b,s_1)$.

3.8 Declarations and performatives

A *declaration* is a speech act that, under appropriate conditions, makes its content true. The content of a declaration must represent an institutional fact, and the agent that makes the declaration has to be authorized to bring about such an institutional fact; besides, it must be ontologically possible to bring about the fact. Consider the act of opening a society's meeting; such an act can be performed by the president of the society by making a suitable declaration. All this presupposes a special Institution of Associations, which in particular will include a definition of meetings in its core ontology, and authorize the president to open a meeting. Additional norms to regulate the exercise of institutional powers can also be defined.

Declarations can be regarded as a universal convention for the performance of all sorts of institutional actions. The logical definition of declarations is particularly interesting, because it states that

(CADecl) $CountAs(declare(\tau),\tau)$.

This definition implies that declaring an action of type τ counts as the actual performance of an action of type τ, provided such a performance is ontologically possible and the actor is authorized to perform actions of type τ, as stated below:

$$Done(e,x,declare(\tau)) \wedge Auth\ (x,\tau) \wedge OntPoss(\tau) \rightarrow Done(e,x,\tau).$$

It is a remarkable fact that, once declarations are defined, it becomes possible to realize all communicative acts introduced so far as declarations. Suppose for example that agent x exchanges with agent y a specific message of type *declare* whose body, expressed as a suitable content language sentence, means "I commit to the truth of 'it is raining' ". The formula

$$Done(e,x,declare(mc(x,y,\text{'it is raining'})))$$

represents such a message exchange. We derive

$$Done(e,x,mc(x,y,\text{'it is raining'}))$$

from our premises and a number of axioms, and then, by Axiom MC, we have that

$$Comm(e,x,y,\text{'it is raining'})$$

holds.

We conclude that the exchange of such a declaration message has the same effect as the exchange of a message of type *inform*. Thus, messages of type *inform* are not strictly necessary, because the same result can be obtained by a declaration, performed in the context of the Basic Institution. If we apply the same line of reasoning to all types of messages, it turns out that all communicative acts can be realized through the use of a single type of messages, namely declaration messages. Carrying out a communicative act by declaration corresponds to the *performative execution* of the communicative act [21]. If all communicative acts that boil down to commitment-manipulation action can be carried out in performative form, then it is possible to define a full ACL starting from one single type of messages, that is, declaration messages.

4 The Operational Specification

We now give an operational specification of the concepts we have modeled in logic in the previous section. To keep close to actual agent programming practice, we shall rely on an object-oriented paradigm to define communicative acts as institutional actions.

As we have already seen, the main components of an institution are the *core ontology*, the set of *authorizations*, the set of *conventions*, and the set of *norms*. In the Basic Institution the core ontology consists of the ontology of commitments; the authorizations specify which agents are empowered to perform actions on commitments; the conventions relate the form of messages with the institutional effects achieved by sending them. No norms are specified by the Basic Institution.

4.1 Commitment

We start from the notion of commitment, that we regard as an object, that is, as a structure with data fields and methods to access and manipulate their values. A commitment has fields for an *identifier*, a *debtor*, a *creditor*, a *content*, a *state*, used to keep track of the temporal evolution of the commitment, and a *timeout*, which is relevant only in some cases and will therefore be treated as an optional parameter. Precommitments are not regarded as a different type of objects, but as one of the possible states in the evolution of a commitment. More precisely, a commitment can be in one of the following states: *unset* (corresponding to a precommitment), *pending* (when the truth value of the content is still undefined), *fulfilled* (when the content is true), *violated* (when the content is false), and *cancelled* (when it no longer exists).

In the logical model, the use of a very expressive temporal logic makes it possible to represent a wide range of content types, like for example *one-shot conditional commitments* (i.e., commitments that bind the debtor only once, when the antecedent of a conditional becomes true) or *standing conditional commitments* (i.e., commitments that bind the debtor whenever the antecedent of a conditional becomes true). In the operational model, it would be unrealistic to assume that actual agents have sufficient logical capacities to process such an expressive language. In some previous works [10],[11] we solved this problem by introducing an additional field for the condition of the commitment. In the model presented in this paper, on the contrary, we prefer to avoid this addition, in order to keep closer to the logical model. As we shall see, the solution proposed in this paper is also more general in that, contrary to our previous proposal, it allows for the representation of standing conditional commitments. Commitment objects will be represented with the following notation:

$$C_{id}(state, debtor, creditor, content[, timeout])$$

where the brackets indicate that the *timeout* parameter is optional.

The content of a commitment is a *temporal proposition* (see Section 4.2). The state of a commitment undergoes a life cycle, described by the statechart diagram in Figure 1. The state of a commitment can change as an effect of the invocation of its basic methods (solid lines) or of environmental events (dotted lines). Relevant events are due to the change of the truth-value of the commitment's content or to the fact that the timeout of an unset commitment has elapsed. We assume that when a commitment object is declared, the constructor of the class creates an empty commitment object, $C_i()$. We represent the invocation of a method by the name of the object followed by a dot and by the name of the method with its parameter list. Commitments are created and manipulated through the following basic operations:

– *Make commitment.* By invoking the method $mc(a, b, P)$ with arbitrary debtor a, creditor b, and content P, a new unset commitment object is created:

  $$C_i().mc(a, b, P[, to]) \rightarrow C_i(unset, a, b, P[, to])$$

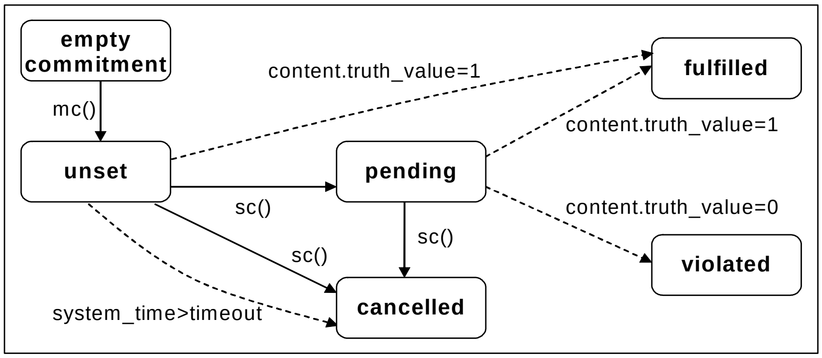

Fig. 1. The life-cycle of commitments.

– *Set commitment.* The method $sc(s)$ changes the current state of an existing commitment object to s:

$$C_i(-, a, b, P).sc(s) \rightarrow C_i(s, a, b, P)$$

There are some ontological presuppositions for the manipulation of the state of commitment objects: first of all the commitment object must exist, second the allowed state changes are only the ones reported in Figure 1.

4.2 The representation of content

As we have seen in Section 3, the definition of an ACL is strictly related to the specification of a content language used to express the content of messages. FIPA, the Foundation for Intelligent Physical Agents, has proposed several CLs [22] (FIPA SL, FIPA CCL, FIPA KIF, FIPA RDF), but none of these provides for a standard treatment of temporal aspects. We think that an application-independent treatment of time is crucial for any practically usable CL; this is especially true if the ACL semantics is defined in terms of commitments, because commitments often specify deadlines for the execution of actions. Therefore, even if we do not intend to define a new content language for ACLs, in this section we propose a representation of content that explicitly takes time into account.

We represent the content of ACL messages, and thus of commitments, as a type of objects that we call *temporal proposition objects*, or *temporal propositions* for short. Every temporal proposition includes a *truth value* that is continually updated by a method, that we call a *truth manager*. As remarked in the DAML Ontology of Time [23], a temporal entity may be an *instant* or an *interval*. Accordingly, our model has *instant propositions* (i.e., temporal proposition objects whose truth value depends on a single instant of time), and *interval propositions* (i.e., temporal proposition objects whose truth value depends on a whole time interval). In turn, an interval proposition can be true if a given state of affairs holds for *every instant* in the associated time interval, or for *some instant* in the associated time interval. To model this aspect, interval propositions have a *mode* attribute, whose value can be either *for all* (∀) or *exists* (∃). Here are a few examples of natural language statements that may be represented as temporal propositions:

- "it is the end of the month" may be represented as an instant proposition, true at all instants belonging to the last day of every month and false otherwise;
- "the service will be accessible for the whole year 2004" may be represented as an interval proposition, whose interval is year 2004 and whose mode is *for all*;
- "the product will be delivered within the current month" may be represented as an interval proposition, whose interval is the current month and whose mode is *exists*;

Besides the truth manager, in charge of updating the truth value, we assume that every temporal proposition has a method, that we call *truth notifier*, in charge notifying every change of the truth value, together with the system time at which such a change took place, to all objects which are "observing" the temporal proposition. We now give a detailed definition of temporal propositions.

Instant propositions An instant proposition $(p, q, ...)$ is an object with fields for:

- the *truth value* of the object, either true (1) or false (0), at the current system time;
- the *time of change*, that is, the most recent time of change of the truth value from 0 to 1 or vice versa.

To represent the semantics of instant propositions we follow an approach akin to Harnad's *symbol grounding* [24]. More specifically, the semantics of an instant proposition is embedded in its truth manager. For example, instant proposition p represents the English sentence "it is the end of the month" if, and only if, p's truth manager keeps p's truth value to 1 during the last day of every month, and to 0 during all other days. This proposition may be viewed as representing an atomic statement, in that its semantics corresponds to the English sentence "it is the end of the month".

It is also possible to build instant propositions that correspond to complex statements, and in particular to Boolean combinations of atomic statements. For example, an instant proposition representing the sentence "it is the end of January" can be built as a Boolean combination of two instant propositions, respectively representing the atomic sentences "it is the end of the month" and "it is January". We assume that for every Boolean connective we have a corresponding class, whose instances are complex instant propositions. For example, an *and* instant proposition is an object that has fields for:

- the *list of components*, which is a list of instant propositions;
- the *truth value* of the object, either 1 or 0, at the current system time;
- the *time of change*, that is, the most recent time of change of the truth value from 0 to 1 or vice-versa.

When an *and* instant proposition p is created, every instant proposition q_i belonging to p's list of components is created with its truth manager, and with a truth notifier that sends a suitable notice to p every time q_i's truth value is updated. Then, p's truth manager will update p's truth value by computing the Boolean conjunction of the truth values of all q_i's. Analogous definitions may be given for all Boolean connectives.

In our operational specification, instant propositions are the starting point for all propositional representations within the system. This assumption is coherent with the prescriptions of the logical model (see Section 3), which is based on the idea that all atomic statements are atemporal (i.e., indexical on the current time instant). In view of our general aims, it is appropriate to assume that an atomic instant proposition may represent:

- a state of affairs, for example: "the price of the product is 100 euros" or "it is the last day of the current month;"
- the execution of an action, either a communicative act defined in a suitable library (see Section 4.3) or an application-dependent action (e.g., "a payment of 100 euros has been made");
- a commitment with given attributes (see Section 4.1).

Simple interval propositions A simple interval proposition is an interval proposition whose truth value depends on an instant proposition, a time interval, and a mode.

A time interval may go from a single instant to the entire life of the system, and is represented by means of its boundaries (written in brackets). In turn, the boundaries of a time interval can be:

- a fixed instant of the system time, represented by a constant numerical value;
- *now*, that is, a reference to the current time instant, typically initialized with the execution time of a communicative act;
- the time of true of an instant or interval proposition (see below);
- an arithmetic expression involving the items above.

A simple interval proposition, $(P, Q, ...)$, is an object with fields for:

- a *statement*, represented by an instant proposition (either atomic or complex);
- a *time interval*;
- a temporal *mode*, either for all ($\forall$) or exist ($\exists$), which specifies whether the statement should be true for the whole time interval or on at least one instant of the time interval;
- a *truth value*, which may be true (1), false (0) or undefined ($\perp$);

– the *time of change* of the truth value from $\perp$ to 1 or to 0.

Like instant propositions, simple interval propositions have a truth manager and a truth notifier. When it is necessary to show their components, such objects will be represented with the following notation:

$$P(statement, time\ interval, mode, truth\ value, time\ of\ change).$$

A major difference between instant and interval propositions is that while the former can only be true or false, the latter can also be undefined. This important fact has to do with the branching structure of time (see Section 3). Consider for example the English sentence “the payment will be made within three days”. This sentence can be represented by the simple interval proposition

$$P(p, [now, now + 3days], \exists, truth\ value, time\ of\ change),$$

where in turn statement p is an instant proposition whose truth manager monitors the execution of the relevant payment and sets p's truth value to true as soon as this is the case. The truth value of P is initialized to $\perp$, and then set to either 1 or 0 according to the rules described below. In particular, P's truth manager will be notified by p as soon as p's truth value is set to true.

In our operational model we assume that when a simple interval proposition is created, its truth value is initialized to $\perp$ and then updated by the truth manager according to the following specifications:

– if the mode is '$\forall$', the truth value is set to 0 if the statement is false at any point of the time interval, otherwise it is set to 1 when the time interval expires;
– if the mode is '$\exists$', the truth value is set to 1 if the statement is true at any point of the time interval, otherwise it is set to 0 when the time interval expires.

These rules specify the operational semantics of simple interval propositions coherently with the truth conditions of sentences as defined in Section 3.3. In particular, the specification implies that the truth value of such an object is monotonic in time, that is, it can switch from $\perp$ to 1 or to 0, and then cannot change any more.

Interval propositions in general Interval propositions are used to represent the content of commitments. Like with simple interval propositions, the truth value of an interval proposition is initialized to $\perp$, can switch from $\perp$ to 1 or to 0, but then cannot switch back to $\perp$. This property is important to guarantee that pending commitments will eventually become fulfilled or violated, and that fulfilled and violated commitments will not change state: indeed, coherently with the logical model (see Section 3.3) a commitment is pending, fulfilled, or violated exactly when its content is respectively undefined, true, or false.

The simplest example of an interval proposition is a simple interval proposition. However, agents need to make commitments also to complex logical combinations of interval propositions. Boolean combinations do not raise difficulties, provided we extend the truth tables to deal with $\perp$. On the basis of the definitions given in the Appendix, it can easily be shown that the truth value of a Boolean combination of interval propositions is monotonic in time. Therefore, if we apply to interval propositions the same construction we have introduced for complex instant proposition, we obtain a class of temporal proposition that behaves monotonically in time, as required for interval propositions.

Boolean combinations, however, are not always sufficient: for example, the Boolean conditional connective is not suitable for an effective operational representation of conditional commitments. To see why this is the case, consider the following examples:

- Suppose an offer is made relative to a single specified commodity (e.g., an apartment), to the extent that the commodity will be transferred to a buyer within one week from the payment of the due price (e.g., 100,000 euros). Moreover, the offer is valid for the whole year 2004. This is an example of a *one-shot conditional commitment*, which can be described by the English sentence: "for all 2004, as soon as a payment for the commodity is made, the commodity is transferred to the buyer within one week";
- Suppose an offer is made relative to products of a specified type (e.g., a cellphone), to the extent that a product of the given type will be transferred to every buyer within one week from the payment of the due price (e.g., 100 euros). Moreover, the offer is valid for the whole year 2004. This is an example of a *standing conditional commitment*, which can be described by the sentence: "for all 2004, every time a payment for a product of the specified type is made, then a product of the specified type is transferred to the buyer within one week".

Both examples involve commitments whose content is best viewed as an event-driven structure. In the first case, the event described by the interval proposition "as soon as a payment for the commodity is made in year 2004" acts as a trigger that generates the interval proposition "the commodity is transferred to the buyer within one week". In the latter case, every event described by the instant proposition "a payment for a product of the specified type is made in year 2004" generates a new interval proposition "a product of the specified type is transferred to the buyer within one week". To express such contents, we introduce two event-driven propositional structures: *on* and *whenever*.

An *on proposition*, used to express one-shot conditional commitments, is an object with fields for:

- the *on condition*, which is an interval proposition, Q;
- the *statement*, which is an interval proposition, P;
- the *truth value*, representing the global truth value of the *on* proposition;
- the *time of change*, representing the time of change of the *on* proposition from $\perp$ to 1 or to 0.

For the sake of simplicity, in our examples (see Section 4.4) an *on* proposition will be represented by the expression P *on* Q.

Every *on* proposition has a truth manager that computes the truth value as follows:

- the interval proposition Q is created, with $Q.truth\_value = \perp$, and its truth manager is set up;
- as soon as $Q.truth\_value = 0$, the truth value of the *on* proposition is set to 1;
- as soon as $Q.truth\_value = 1$, the interval proposition P is created with $P.truth\_value = \perp$, and its truth manager is set up (the boundaries of P's interval may depend on Q's time of change). The truth value of the *on* proposition is then given by $P.truth\_value$.

To express interesting one-shot conditional commitments, it is important that the boundaries of the statement's time interval may depend on the *time of change* of the *on* condition. An example of a conditional commitment of this type is reported in Section 4.4.

A *whenever* proposition, used to express standing conditional commitments, is an object with fields for:

- the *whenever* condition, which is an instant proposition, q;
- the *reference interval*, which is the time interval, $[t_{start}, t_{end}]$, in which the truth value of the *whenever* condition has to be monitored;
- the *statement*, which is an interval proposition P;
- the *truth value*, representing the global truth value of the *whenever* proposition;
- the *time of change*, representing the time of change of the *whenever* proposition from $\perp$ to 1 or to 0.

In our examples (see Section 4.4), a *whenever* proposition will be represented by the expression P *whenever* q *in* $[t_{start}, t_{end}]$.

Every *whenever* proposition has a truth manager that computes the truth value as follows:

- as soon as the current system time reaches t_{start}, the instant proposition q is created and its truth manager is set up;
- every time that $q.truth\_value = 1$, an interval proposition object P_i is created ($i = 1, 2, ...$), with $P_i.truth\_value = \perp$, and its truth manager is set up (the boundaries of P_i's interval may depend on q's time of change);
- as soon as the current system time reaches t_{end}, the truth value of the *whenever* proposition is set to the Boolean conjunction of all the P_i's that have been created; given that the P_i's are interval proposition objects, it may be necessary to wait for all the corresponding intervals to expire before producing the final truth value.

The truth value of a *whenever* proposition may be computed more efficiently by noting that:

- as soon as one of the P_i's becomes false, the whole *whenever* proposition may be set to false;

- as soon as one of the P_i's becomes true, it may be deleted by the set of conjuncts that defines the truth value of the *whenever* proposition;
- if the set of conjuncts that defines the truth value of the *whenever* proposition becomes empty, the truth value of the *whenever* proposition may be set to true.

To summarize, a representation for the content of a commitment is an interval proposition, which may be:

- a simple interval proposition;
- a Boolean combination of interval propositions;
- an *on* proposition;
- a *whenever* proposition.

This completes the treatment of commitments and of their contents. We now proceed to another important aspect of agent communication, that is, the library of communicative acts.

4.3 A library of Communicative Acts

The elementary operations on commitment (Section 4.1) should not be viewed as actions to be directly performed by agents; rather, they are low-level methods used to implement operations on commitment objects. Agents do not directly invoke these methods, but manipulate commitments through a library of communicative acts (see Section 4.3), according to their institutional powers. Such powers are defined by a set of institution-dependent *authorizations* which, in order to abstract from specific agents, are expressed in terms of *roles*. The Basic Institution has a role for registered agents, *RegAgt*, which is the role that every agent takes when it becomes part of the "society of agents."; moreover, every commitment implicitly defines two roles, *debtor* and *creditor*, which affect the institutional power of agents. Coherently with the prescriptions of the logical model, the authorizations of the Basic Institution are:

- a registered agent may create an *unset* commitment with any registered agent as the debtor and the creditor;
- the debtor of an *unset* commitment can set it to either *pending* or *cancelled*;
- the creditor of an *unset* or *pending* commitment can set it to *cancelled*.

Special institutions may enlarge or modify this basic set of authorizations according to need.

In our operational model, we have not yet defined a general mechanism to guarantee that actions are performed by authorized agents. While this is a topic for future research, for the time being we assume that all communicative acts are defined in a communicative act library in such a way that the authorizations of the Basic Institution are met.

We shall now define the meaning of the basic types of communicative acts as identified by Speech Act Theory [13]. We assume that the agents involved are registered as members of the Basic Institution. With the exception of declarations, that will be dealt with in a special way, all communicative acts defined are compatible with the authorizations of the Basic Institution. As a whole, the library of communicative acts can be regarded as the set of conventions (see Section 3.6) of the Basic Institution.

In the following definitions the symbol ":=" means that the act represented on the left-hand side is actually performed through the invocation of the methods listed on the right-hand side.

Assertives We consider the *inform* act as our prototypical assertive act. This act is used by agent a to inform agent b that proposition P holds. In a commitment-based approach, an act of informing can be defined as follows:

$$inform(a, b, P) := \{C_i().mc(a, b, P); C_i(unset, a, b, P).sc(pending)\}.$$

Directives We treat *request* as our basic directive act, and define it as the creation of an unset commitment with the sender as the creditor and the receiver as the debtor. The request by agent a to agent b to bring about proposition P is defined as:

$$request(a, b, P, to) := \{C_i().mc(b, a, P, to)\}.$$

Questions (or queries) are requests to be informed about something (see [10] for a definition of *yes-no-question* and *wh-question*).

Commissives Here we define the basic commissive act of *promising*. A promise by agent a to agent b to bring about proposition P is defined as:

$$promise(a, b, P) := \{C_i().mc(a, b, P);\ C_i(unset, a, b, P).sc(pending)\}.$$

Two main types of commissive acts can be performed only in connection with an unset commitment, namely *accept*, and *refuse*. These act have nontrivial *ontological possibility* preconditions (see Section 3.2), and can be defined as follows:

$$preconditions:\ \exists\, C_i(unset, b, a, P, to)$$
$$accept(b, a, C_i(unset, b, a, P, to)) := \{C_i(unset, b, a, P, to).sc(pending)\}$$

$$preconditions:\ \exists\, C_i(unset, b, a, P, to)$$
$$refuse(b, a, C_i(unset, b, a, P, to)) := \{C_i(unset, b, a, P, to).sc(cancelled)\}.$$

Declarations The point of a declaration is to bring about a change in the world, obviously not in the physical or natural world but in an institutional world, that is, a conventional world relying on common agreement among the interacting agents [12]. Declarations actually change the institutional world simply in virtue of their successful performance.

Coherently with the logical model, it is necessary to identify what agents are authorized to perform a given declaration. Typically, authorizations are granted to agents in virtue of the role they play in an interaction, and thus authorizations are naturally associated to roles. A complete treatment of declarations will require an operational model of special institutions that, as we have already remarked, is a topic for future research. However, we suggest here a simplified treatment, that we have adopted elsewhere to deal with the special institution of English auctions [11].

Let us assume that the environment in which the agents interact contains a number of *institutional objects*, O_i, that represent aspects of institutional reality. For example, certain institutional objects may represent states of ownership; such objects will at least include two fields, one for the *owner* and one for the *commodity* owned, and a method (*set_owner*) for setting the value of the owner field. In such an environment, institutional actions may be viewed as actions that modify the values of the fields of institutional objects. For example, to represent the action of transferring the ownership of commodity c from agent a to agent b we can use an institutional object O such that $O.commodity = c$; this object will have $O.owner = a$ before ownership is transferred to b, and $O.owner = b$ after such an action is performed. The action itself can be executed by a method call,

$$O(owner = a, commodity = c).set\_owner(b).$$

Like the methods for commitment manipulation, *set_owner* should not be viewed as an action that can be directly performed by agents, but as a low-level procedure used to implement an institutional action. To actually perform such institutional action, an agent needs to perform a suitable communicative act, which will succeed only if the agent has sufficient institutional powers. As it has been clarified in Section 3.8, only one communicative act type is sufficient to perform all kinds of institutional actions, namely *declarations*. However, it is necessary to specify *authorization* preconditions to perform any specific declaration. To do so, we then introduce a construct to express that an agent having a given role has the power to bring about an institutional change of a given kind:

$$preconditions : x\ has\ a\ given\ role$$
$$declare(x, O_j, \text{``}field_k\text{''}, y) := \{O_j.set\_field_k(y)\}.$$

Using a construct of this type it is possible, for example, to state that the owner of a commodity is authorized to transferring its ownership by declaration:

$$preconditions : x = O.owner()$$
$$declare(x, O, \text{``}owner\text{''}, y) := \{O.set\_owner(y)\}.$$

4.4 Examples

Example 1 A common type of interaction is when a *seller* commits to delivering a specific product to a *buyer* within a given *deadline*, and the buyer commits to paying the price of the product to the seller within a given *delay* from its delivery. This situation involves two commitments, which can be described as follows:

- commitment 1: the seller commits, relative to the client, to delivering the specified product within the specified deadline;
- commitment 2: the buyer commits, relative to the seller, to paying the price of the product within the given delay from delivery.

In other words, the seller is unconditionally committed to delivering the product; when the product is actually delivered, the condition of the buyer's commitment becomes true, and then the buyer is unconditionally committed to paying its price. Note that the time interval of this second commitment is defined only after the associated condition becomes true.

To represent the two commitments, let us first define two instant propositions as follows:

- p means that "the product is delivered to the buyer";
- q means that "the product's price is paid to the seller".

We then define two interval propositions as follows:

- P is the interval proposition $P(p, [now, deadline], \exists)$;
- Q is the interval proposition $Q(q, [P.time\_of\_change(), P.time\_of\_change() + delay], \exists)$.

The seller can now make its commitment by performing the communicative act $promise(seller, buyer, P)$, which creates the commitment $C_1(pending, seller, buyer, P)$. In turn, the buyer can make its commitment by performing the communicative act $promise(buyer, seller, Q\ on\ P)$, which creates the commitment $C_2(pending, buyer, seller, Q\ on\ P)$.

Example 2 Consider an *employer* promising to an *employee* that for all year 2004, the employee's salary will be paid within one week from the end of each month. This case is similar to Example 1, but involves a standing conditional commitment.

To represent the commitment, let us first define two instant propositions as follows:

- p means that "it is the end of the current month";
- q means that "the employee's salary is paid".

We then define an interval proposition as follows:

- Q is the interval proposition $Q(q, [p.time\_of\_change(), p.time\_of\_change() + 7days], \exists)$.

We define t_{start} = 00:00/01/01/2004 and t_{end} = 24:00/31/12/2004. The employer can now make its commitment by performing the communicative act $promise(employer, employee, Q\ whenever\ p\ in\ [t_{start}, t_{end}])$, which creates the commitment $C_1(pending, employer, employee, Q\ whenever\ p\ in\ [t_{start}, t_{end}])$.

Example 3 Consider a situation where the *provider* of a service reserves a *resource* for a *client* (for example a room in a hotel, a place on a flight, a table at a restaurant, etc.) for a time interval from t_1 to t_2; more precisely, the provider commits, relative to the client, to giving the reserved resource to the client (within a maximum delay of $delta_1$ time units) if the client *claims* it during $[t_1, t_2]$.

This example is more complex that it may appear. From the point of view of Speech Act Theory, a *claim* is a directive act, and therefore is akin to a request. What distinguishes a claim from a plain request is that it has the additional precondition (or "preparatory condition," in Searle's terminology) that the sender has the right to have its claim accepted. In a commitment-based framework, a right of agent a can be represented in terms of a commitment of some other agent, b, relative to a. In our case, the simplest solution is to assume that the provider is committed to accepting an unset commitment of given form. This solution, however, introduces a further parameter, that is, the maximum delay, δ_2, for accepting the unset commitment (of course, δ_2 must be smaller than δ_1).

More precisely, let:

- p be the instant proposition meaning "the resource is given to the client";
- P be the interval proposition $P(p, [now, now + \delta_1], \exists)$, meaning "the resource is given to the client within δ_1 time units";
- q be the instant proposition meaning "a commitment of form $C(unset, provider, client, P)$ is made" (the time at which this commitment is created will set the *now* parameter in P);
- Q be the interval proposition $Q(q, [t_1, t_2], \exists)$, meaning "a commitment of form $C(unset, provider, client, P)$ is made during the interval from t_1 to t_2";
- r be the instant proposition meaning "the provider accepts the commitment $C(unset, provider, client, P)$";
- R be the interval proposition $R(r, [now, now + \delta_2], \exists)$, meaning "the provider accepts the commitment $C(unset, provider, client, P)$ within δ_2 time units".

The provider can now make its commitment by performing the communicative act $promise(provider, client, R\ on\ Q)$. To make its claim, the client has to create a suitable unset commitment, which can be done by performing the communicative act $request(client, provider, P)$.

5 Conclusions

In this paper we have presented a logical model and an operational specification of the social concepts necessary to specify the semantics of ACLs. Then, using such concepts, we have defined the meaning of a set of basic communicative acts that can be exploited by heterogeneous agents to interact in an open framework.

Our proposal is based on two main assumptions: (i) that language is the fundamental component of every communicative interaction, and (ii) that communication is an institutional activity, in that it is made possible by social institutions that exist thanks to the common agreement of the interacting agents (or, in the case of artificial agents, of their designers)(see [15]). We suggest that the basic components of an institution are: the ontology of a fragment of social reality, a set of authorizations to perform actions within the institution, the conventions to perform such actions, and a (possibly empty) set of norms that regulate the agents' behavior.

In our view, a social concept that is central to the treatment of communication is commitment. Therefore, we have defined what we call the Basic Institution, to define the concepts, authorizations and conventions related to commitment manipulation. We then view most types of communicative acts as conventions for the manipulation of commitments. We also remark that some kinds of communicative acts require more specific institutional settings. However, a detailed analysis of the special institutions required for agent interaction is a topic for future research.

Another aspect that deserves further research is content language, that is, the language used to represent the content of communicative acts. In this paper we have proposed no specific content language, and have adopted two different representations of contents. More specifically, contents have been represented as first-order terms in the context of the logical model, and as object structures in the context of the operational specification. However, we believe that the relationship between our representations and actual content languages has to be better clarified. In particular, if we want ACLs to reach the level of real applications it will be necessary to establish a standard treatment of time in content languages.

Acknowledgments

We thank Francesco Viganò for useful observations on the operational specification.

Appendix

Truth-Tables for Interval Propositions

Truth-tables of Boolean operators of classical propositional logic can be used to obtain the extended truth-tables of the same operators when the propositions can be true (1), false (0), or undefined($\perp$). The key move is to regard the undefined value as "either 0 or 1", that we represent as $\{0, 1\}$.

As an example, the classical truth table of the *and* operator ($\wedge$) is shown in Table 1, and the extended truth table of the same operator is shown in Table 2.

Table 1. Truth table of the "and" operator

$\wedge$	0	1
0	0	0
1	0	1

The extended truth-table is obtained from the classical one by writing {0,1} when it is impossible to write a single truth value. For example {0,1} $\wedge$ 0 gives 0 because the 0 column in Table 1 contains only 0's, while {0,1} $\wedge$ 1 gives $\{0, 1\}$ because the 1 column in the same table has both 0's and 1's.

Table 2. Extended truth table of the "and" operator

$\wedge$	0	1	{0,1}
0	0	0	0
1	0	1	{0,1}
{0,1}	0	{0,1}	{0,1}

References

1. Cohen, P., Levesque, H.: Rational interaction as the basis for communication. In Cohen, P., Morgan, J., Pollack, M., eds.: Intentions in communication. MIT Press (1990) 221–156
2. Finin, T., Labrou, Y., Mayfield, J.: KQML as an agent communication language. In Bradshaw, J., ed.: Software agents. The MIT Press, Cambridge, MA (1995)
3. Foundation for Intelligent Physical Agents: FIPA 97 Specification Part 2: Agent Communication Language (1997)
4. Singh, M.P.: Agent communication languages: Rethinking the principles. IEEE Computer **31** (1998) 40–47
5. Singh, M.P.: A social semantics for agent communication languages. In: Proceedings of IJCAI-99 Workshop on Agent Communication Languages. (1999) 75–88
6. Colombetti, M.: A commitment–based approach to agent speech acts and conversations. In: Proc. Workshop on Agent Languages and Communication Policies, 4th International Conference on Autonomous Agents (Agents 2000), Barcelona, Spain (2000) 21–29
7. Pitt, J., Mamdani, A.: A protocol-based semantics for an agent communication language. In Thomas, D., ed.: Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99-Vol1), Morgan Kaufmann Publishers (1999) 486–491
8. Jones, A., Parent, X.: Conventional signalling acts and conversation. In Dignum, F., ed.: Advances in Agent Communication. LNAI. Springer Verlag (to be published)
9. Verdicchio, M., Colombetti, M.: A logical model of social commitment for agent communication. In Rosenschein, J.S., Sandholm, T., Wooldridge, M., Yokoo, M., eds.: Proc. Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2003), Melbourne, Australia, ACM Press (2003) 528–535
10. Fornara, N., Colombetti, M.: Operational specification of a commitment-based agent communication language. In Castelfranchi, C., Johnson, W.L., eds.: Proc. First International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2002), Bologna, Italy, ACM Press (2002) 535–542

11. Fornara, N., Colombetti, M.: Defining interaction protocols using a commitment-based agent communication language. In Rosenschein, J.S., Sandholm, T., Wooldridge, M., Yokoo, M., eds.: Proc. Second International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2003), Melbourne, Australia, ACM Press (2003) 520–527
12. Colombetti, M., Verdicchio, M.: An analysis of agent speech acts as institutional actions. In Castelfranchi, C., Johnson, W.L., eds.: Proc. First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2002), Bologna, Italy, ACM Press (2002) 1157–1166
13. Searle, J.R.: Speech Acts: An Essay in the Philosophy of Language. Cambridge University Press, Cambridge, United Kingdom (1969)
14. Alberti, M., Gavanelli, M., Lamma, E., Mello, P., Torroni, P.: Modeling interactions using social integrity constraints: a resource sharing case study. LNAI, Springer Verlag (2004) in this volume
15. Searle, J.R.: The construction of social reality. Free Press, New York (1995)
16. Jones, A., Sergot, M.J.: A formal characterisation of institutionalised power. Journal of the IGPL **4** (1996) 429–445
17. Emerson, E.A., Halpern, J.Y.: ‘Sometimes’ and ‘Not Never’ Revisited. Journal of the ACM **33** (1986) 151–178
18. Esteva, M., Rodríguez-Aguilar, J.A., Sierra, C., Garcia, P., Arcos, J.L.: On the formal specification of electronic institutions. In Dignum, F., Sierra, C., eds.: Agent Mediated Electronic Commerce, The European AgentLink Perspective (LNAI 1991). Springer (2001) 126–147
19. Vasconcelos, W.W.: Logic-based electronic institutions. LNAI, Springer Verlag (2004) in this volume
20. Reichenbach, H.: Elements of Symbolic Logic. MacMillan, New York (1947)
21. Searle, J.R., Vanderveken, D.: Foundations of illocutionary logic. Cambridge University Press, Cambridge, UK (1985)
22. Foundation for Intelligent Physical Agents: FIPA Content Language (CL) Specifications. http://www.fipa.org (2003)
23. Ferguson, G., Allen, J., Fikes, R., Hayes, P., McDermott, D., Niles, I., Pease, A., Tate, A., Tyson, M., Waldinger, R.: A DAML ontology of time. http://www.cs.rochester.edu/ ferguson/daml/ (2002)
24. Harnad, S.: The symbol grounding problem. Physica D **42** (1990) 335–346