

ITC

Institute for
Communication
Technologies

www.itc.com.unisi.ch

Faculty of
Communication
Sciences

Università della
Svizzera
italiana

via Buffi 13
CH-6900
Lugano

Technical report

No. 1

Agent communication and institutional reality (extended version)

Nicoletta Fornara, Francesco Viganò, and Marco Colombetti

Institute for Communication Technologies
Università della Svizzera Italiana, 2004.

Agent Communication and Institutional Reality^{*}

Nicoletta Fornara¹, Francesco Viganò¹, and Marco Colombetti^{1,2}

¹ Università della Svizzera italiana, via G. Buffi 13, 6900 Lugano, Switzerland
{nicoletta.fornara, francesco.vigano, marco.colombetti}@lu.unisi.ch,

² Politecnico di Milano, piazza Leonardo Da Vinci 32, Milano, Italy
marco.colombetti@polimi.it

Abstract. In this paper we propose to regard an Agent Communication Language (ACL) as a set of conventions to act on a fragment of institutional reality, defined in the context of an artificial institution. Within such an approach, we first reformulate a previously proposed commitment-based semantics for ACLs. In particular we show that all commonly used types of communicative acts can be defined in terms of a single basic type, namely *declarations*, within an artificial institution that we call Basic Institution. We then go on defining special institutions, that augment the Basic Institution by adding ontological and normative elements. Finally, as an example of a special institution we give a partial definition of the institution of English Auctions.

1 Introduction

In the last few years the concept of *social commitment* has been largely used by a growing number of researchers to define the semantics of Agent Communication Languages (ACLs). After the first studies carried out by Singh and by Colombetti [24, 6], further investigations have been carried out from an operational point of view [12, 18], following a logical approach [26], and in the field of argumentation studies [1, 3]. The main advantages of this approach are that commitments are objective and independent of an agent's internal structure, and that it is possible to verify whether an agent is behaving according to the given semantics.

Social commitments are used to represent the evolution of social relationships among agents during interactions. Communicative acts are then viewed as actions carried out to modify such relationships by creating, updating or cancelling commitments according to a predefined set of shared rules [26, 13]. More precisely, communicative acts are regarded as a sort of *institutional actions*, that is, as actions performed within an *institution* to modify a fragment of social reality [22]. Defining the semantics of an ACL has therefore two sides: one side is the definition of the institutional effects brought about by the performance of communicative acts; the other side is the definition of the social context in which agents can carry out institutional actions, and that we call an (*artificial*) *institution*. Indeed, our main tenet is that without the definition of an appropriate institution it is impossible to specify the semantics of an ACL.

^{*} Supported by Swiss National Science Foundation project 200021-100260, "An Open Interaction Framework for Communicative Agents"

This paper is structured as follows. In Section 2 we introduce the fundamental concepts on which we base our treatment of agent communication, namely the concepts of an *institutional action*, of a *convention*, and of a "counts as" relationship between an *instrumental action* and the corresponding communicative act. In Section 3 we define the institutional actions that can be performed on commitments. In Section 4 we describe the Basic Institution (i.e., the institution that regulates the management of commitments) and introduce the concept of a special institution. In Section 5 we give a partial description of a specific case of a special institution, that is, the institution of English Auctions. Finally in Section 6 we draw some conclusions and delineate some directions for future work.

2 Fundamental concepts

We view a multiagent system (MAS) as a technological extension of human society, by which individual persons and human organizations can delegate the execution of institutional actions to the artificial system. Examples of such actions are establishing appointments, signing contracts, and carrying out commercial transactions. For this reason there are strong connections between some aspects of a MAS and some aspects of human society, and therefore the concepts used to model a MAS interaction framework have to reflect some crucial characteristics of their human counterpart.

The context within which artificial agents operate can be modelled as consisting of a set of *entities* that can have *natural* or *institutional* attributes, that is, attributes that exist only thanks to the common agreement of the interacting agents (or more precisely of their users). For example, the color of a book is a natural attribute, while the book's price and its owner are institutional attributes. Natural attributes are assumed to reflect the physical properties of the corresponding entities of the real world, and typically cannot be changed by artificial agents (unless the agent controls a physical robot). On the contrary, institutional attributes can be affected by *institutional actions* performed by purely software agents.

2.1 Institutional actions

Institutional actions are particular types of actions [8] that are crucial for the formalization of communicative interactions taking place in open interaction frameworks. The effect of institutional actions is to change institutional attributes, that exist only thanks to common agreement. Therefore, agents cannot perform such actions by exploiting causal links occurring in the natural world, as it would be done to open a door or to remove a physical object. Rather, as we shall see, institutional actions are performed on the basis of a shared set of conventions.

Because of their intrinsic social nature, a crucial condition for the actual performance of institutional actions is that they must be *public*, that is, made known to the relevant agents by means of some action that can be directly executed by

an artificial agent. It is therefore natural to assume that all institutional actions are performed by sending suitable messages to the relevant agents. An example of institutional action, that will be discussed in Section 5, is the act of opening an auction; as we shall see, an agent (the auctioneer) can perform such an action by sending a suitable message to the relevant group of agents (the participants). However, the act of sending the message is merely instrumental, and should not be confused with the institutional action of opening the auction.

We define institutional actions by specifying their *preconditions* and *postconditions*, therefore abstracting from the way in which such actions are concretely carried out. More precisely, an institutional action is characterized by:

- an *action name* followed by a possibly empty list of *parameters*;
- a possibly empty set of (*ontological*) *preconditions*, that specify the values that certain institutional attributes must have for the action to be meaningful (for example, opening an auction is meaningful only if the auction is not already open);
- a nonempty set of *postconditions*, that specify the values of certain institutional attributes after a successful performance of the action.

2.2 Instrumental actions

As we have already remarked, an institutional action is performed by executing an instrumental action, conventionally associated to the institutional action. In the human world such instrumental actions vary from certain bodily movements (raising one’s arm to vote), to the use of specific physical tools (waving a white flag to surrender), to the use of language (saying ”the auction is open” to open an auction). In a system of artificial agents, it is natural to assume that all institutional actions are performed by means of a single type of instrumental actions, namely exchanging a message.

For the purposes of the current treatment, a message consists of: a *message type*, a *sender*, one or more *receivers*, and a *content*. The action of exchanging a message will be represented with the following notation:

`exchMsg(message.type, sender, receiver(s), content)`

Note that here *sender* and *receiver* are just fields of a message. That such fields correctly represent the agent that actually sends the message and the agents to which the message is delivered has to be guaranteed by the underlying message transport system.

2.3 The ”counts as” relation

Following Searle [22], the construction of social reality in the human world is possible thanks to *constitutive rules* of the form *X counts as Y in C*; in the particular case where *X* and *Y* are actions, the performance of an action of type *X* in context *C* can count as performing an action of type *Y*. Similarly, in an artificial system, thanks to shared *conventions*, the action of exchanging a

particular message can "count as" the execution of some institutional action, if certain *contextual conditions* are satisfied.

According to Searle's Speech Act Theory [21], *declarations* are the particular category of communicative acts whose point is to bring about a change in the institutional reality in virtue of their successful performance. By definition the content of a declaration describes precisely the institutional changes that it brings about. Therefore, we take messages of type *declare* as the fundamental means to perform institutional actions. The convention that binds the exchange of a *declare* message to the performance of the institutional action (*iaction*), described in its content, can be described as follows:

$$\text{exchMsg}(\text{declare}, \text{sender}, \text{receiver}, \text{iaction}(\text{parameters})) \underset{\text{conv}}{=} \text{iaction}(\text{parameters})$$

By itself, however, a convention is not sufficient to guarantee the successful performance of an institutional action by the exchange of a *declare* message: indeed, some additional conditions about the agent that sends the message and about the state of system must be satisfied. In general, an agent must be *authorized* to perform an institutional action; for example, only the auctioneer can open an auction by sending a suitable message to the participants. Further *contextual conditions* about the state of the system, expressed by suitable Boolean expressions, may be required; for example, it may be established that an auction is validly opened only if there are at least two participants.

Assuming that every agent in the interaction system has an identifier (*agent_id*), authorizations will be represented with the following notation:

$$\text{Auth}(\text{agent_id}, \text{iaction}(\text{parameters}), \text{contextual_conditions})$$

Our notion of authorization should not be confused with the notion of *permission*. The distinction we make between these two concepts is similar to the one between institutionalized power and permission proposed by Jones and Sergot in [16]. While authorizations are necessary conditions for the performance of institutional actions, permissions (like obligations) are brought about by *norms* (see Section 4.2), that is, by rules that affect the normative positions of the agents in the system. The crucial difference between authorizations and permissions is highlighted in the cases when they are not granted. If an agent is not authorized to perform an institutional action, a performance of the corresponding instrumental action does not count as a performance of the institutional action (the institutional action is thus not executed). On the contrary, if an authorized agent performs an institutional action without permission, the institutional action is successfully performed, but the agent violates a norm and it may be sanctioned for its behavior.

In the specification of an interaction system it is useful to express authorizations in term of the *roles* filled by agents, in order to abstract from the concrete agents that will be actually involved in an interaction. For example, the authorization to open and close an auction is granted to the agent that fills the role of the auctioneer, independently of its individual identity.

The concept of a role is very broad: for example, it is possible to regard social commitments as institutional entities that define two roles: the *debtor* of the commitment and its *creditor*. This fact appears to be general; that is, roles are defined relative to an institutional entity. We can then abstractly define the authorization to perform a specific institutional action (with given parameters) associating it to a role defined in the context of a specific institutional entity (*ientity*):

$$\text{Auth}(\text{ientity.role}, \text{iaction}(\text{parameters}), \text{contextual_conditions})$$

In a concrete interaction, the authorizations associated to roles need to be transformed into authorizations of an actual agent in the system. Such transformation can be obtained searching among all the institutional entities present in the system the ones that match the description given through the parameters of the institutional action, and then creating a concrete authorization for each agent having the role indicated in the abstract authorization.

3 A commitment-based Agent Communication Language

The semantics of ACLs that we have proposed in [12, 13] is based on the assumption that the performance of a communicative act in a multiagent system has the effect of changing the social relationship between the sender and the receiver, and that this change can be represented by means of an institutional entity, that is, *social commitment*. To specify the meaning of various types of communicative acts in terms of effects on commitments, it is necessary to define an ontology of commitment and the institutional actions necessary to operate on commitments.

3.1 The Ontology of Commitment

We regard a commitment as an entity with the following attributes: an *identifier*; a *debtor*; a *creditor*; a *content*; a *state*, used to keep track of the temporal evolution of the commitment. Commitments will be represented with the following notation:

$$\text{Comm}_{id}(\text{state}, \text{debtor}, \text{creditor}, \text{content})$$

The content of a commitment can be represented by means of a *temporal proposition* (for a detailed treatment of temporal propositions see [13, 7]), that is, a proposition about a state of affairs or about the performance of an action, referred to a specific interval of time. At every time instant, a temporal proposition has a truth value, that can be *undefined*, *true*, or *false*.

The state of a commitment undergoes a life cycle, described by the state diagram of Figure 1, and can change as an effect of the execution of institutional actions (solid lines) or of environmental events (dotted lines). Relevant events are due to the change of the truth-value of the commitment's content.

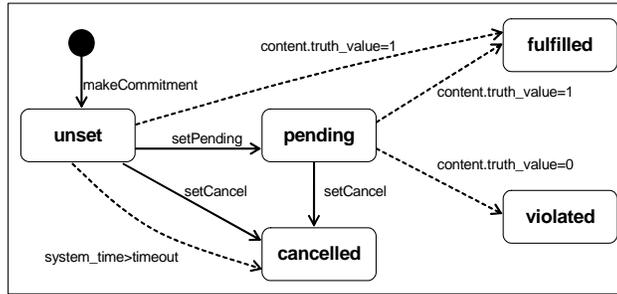


Fig. 1. The life-cycle of commitments.

The creditor of a commitment can be a single agent or a group of agents. It is important to remark that a commitment taken with a group of agents need not be equivalent to a conjunction of commitments taken with every member of the group. This point has been thoroughly analyzed in the literature [5, 9] but is behind the scope of this paper.

Institutional actions on commitment The institutional actions that operate on commitments are defined below; preconditions and effects are described using Object Constraint Language (OCL) [20].

- name : `makeCommitment(debtor, creditor, content)`
 pre : `not Comm.allInstances → exists(c|c.debtor = debtor and c.creditor = creditor and c.content = content)`
 post : `Comm.allInstances → exists(c|c.state = unset and c.debtor = debtor and c.creditor = creditor and c.content = content)`
- name : `setCancel(debtor, creditor, content)`
 pre : `Comm.allInstances → exists(c|(c.state = unset or c.state = pending) and c.debtor = debtor and c.creditor = creditor and c.content = content)`
 post : `Comm.allInstances → exists(c|c.state = cancel and c.debtor = debtor and c.creditor = creditor and c.content = content)`
- name : `setPending(debtor, creditor, content)`
 pre : `Comm.allInstances → exists(c|c.state = unset and c.debtor = debtor and c.creditor = creditor and c.content = content)`
 post : `Comm.allInstances → exists(c|c.state = pending and c.debtor = debtor and c.creditor = creditor and c.content = content)`

It is often useful to define *institutional macro-actions*, that is, actions whose execution coincides with the sequential execution of a list of existing institutional actions, conceived of as a single transaction. For example:

name: `makePendingComm(debtor, creditor, content) =def makeCommitment(debtor, creditor, content), setPending(debtor, creditor, content)`

3.2 Communicative Acts Libraries

As we already discussed in Section 2, the exchange of a message of type `declare` can be considered as the universal act for the performance of institutional actions; in particular, every type of communicative act can be performed by means of a declaration. This means that, at least in principle, an ACL can be defined on the basis of a single type of messages¹.

To make a more natural set of communicative acts available to human developers, we now define a library of messages that gets closer to FIPA ACL [15]. The content of all messages defined below is a temporal proposition, that is, a description of a state of affairs or the description of a physical action referred to a certain interval of time. The symbol $=_{def}$ means that performing the action on the left-hand side is the same as performing the action on the right-hand side.

- `exchMsg(inform, sender, receiver, content) =def exchMsg(declare, sender, receiver, makePendingComm(sender, receiver, content))`
- `exchMsg(request, sender, receiver, content) =def exchMsg(declare, sender, receiver, makeCommitment(sender, receiver, content))`
- `exchMsg(accept, sender, receiver, content) =def exchMsg(declare, sender, receiver, setPending(sender, receiver, content))`
- `exchMsg(reject, sender, receiver, content) =def exchMsg(declare, sender, receiver, setCancel(sender, receiver, content))`

4 Artificial Institutions

The word *institution* is used in the literature with different meanings. An institution can be seen as an established organization (especially of a public character) with a code of law, like for example a hospital or a university. With a different meaning, the word is used to refer to a set of concepts that exist only thanks to the common agreement of a community of agents, like for example in the case of money, ownership, or marriage.

In multiagent systems research the term *artificial institution* is commonly used to refer to a specific organization or to an abstract pattern that regulates the interaction among agents [10] [25]. On the contrary, we use the term “artificial institution” to refer to the abstract description of shared concepts and rules that regulate a fragment of social reality. In this perspective a concrete organization is a reification of one or more artificial institutions. In our view, the specification of an institution consists of the following components:

- the *core ontology*, that is, the definitions of the institutional concepts introduced by the institution and of the institutional actions that operates on them;

¹ Carrying out a communicative act by declaration corresponds to a *performative execution* of the communicative act [23]. In human languages, however, only the communicative acts that are completely overt may have a performative execution; certain communicative acts, like for example the act of insinuating, cannot be performed by declaration, because they intrinsically contain a concealed component.

- a set of *authorizations* specifying which agents are empowered to perform the institutional actions;
- a set of *norms* that impose *obligations* and *permissions* on the agents that interact within the institution.

Of course, in order that the proposed model can actually be used in real applications it is necessary that the fundamental concepts, used to define the structure of institutions, are collectively accepted by the designers and users of open interaction frameworks.

4.1 The Basic Institution

The Basic Institution is the institution that defines and regulates the management of commitments, which we regard as the fundamental concept of every interaction. In the previous section commitment has been introduced as an institutional entity, together with a set of institutional actions to operate on it. We showed that commitments can be used to define basic types of communicative acts that can be performed by exchanging *declare* messages.

As discussed in Section 2, the "count as" relation between the action of exchanging a message and the associated institutional action takes place if some conditions are satisfied; more precisely, the sender of the message must be authorized to perform the institutional action and some contextual conditions must hold. We have also shown how authorizations can be associated to roles.

We now define a set of authorizations concerning the creation and the manipulation of commitments. Such authorizations will be associated to the two roles introduced by commitments themselves: the role of *debtor* and the role of *creditor*. Moreover, we assume a universal role, *RegAgt*, that every registered agent plays throughout its lifetime.

- Any registered agent can create an *unset* commitment with any other registered agent as debtor or creditor:

Auth(RegAgt, makeCommitment(*debtor*, *creditor*, *content*));

- the debtor of an *unset* commitment can set it to *pending*:

Auth(Comm(*debtor*, *creditor*, *content*).debtor, setPending(*debtor*, *creditor*, *content*));

- the debtor of an *unset* commitment can set it to *cancelled*:

Auth(Comm(unset, *debtor*, *creditor*, *content*).debtor, setCancel(*debtor*, *creditor*, *content*));

- the creditor of a commitment can set it to *cancelled*:

Auth(Comm(*debtor*, *creditor*, *content*).creditor, setCancel(*debtor*, *creditor*, *content*)).

Note that these authorizations allow an agent to perform all communicative acts defined in Section 3.2. These basic authorizations may be modified or new ones may be introduced within special institutions (see Section 5).

In general, institutions also define sets of norms to regulate the behavior of agents. In our current view, the Basic Institution specifies no norms. However, norms are introduced by most special institutions, and in particular by the special institution of English Auctions described in Section 5. Therefore, in the next subsection we give a detailed description of our concept of norm.

4.2 Norms

In a special institution, the execution of an action by an authorized agent often needs to be regulated by another fundamental component of artificial institutions, that is, a system of *norms*. For example, the auctioneer of an English Auction not only is authorized to declare who is the winner, but he is also obliged to do so in certain circumstances. Furthermore, there are conditions under which it is forbidden to the auctioneer to declare an agent as the winner (for instance during a period of time reserved for offers).

Norms prescribe which institutional actions should or should not be executed among those that are authorized. In doing so, norms play an important function, in that they make an agent's behavior at least partially predictable and allow agents to coordinate and plan their actions according to the expected behavior of the others, as studied in [19, 2]. In particular, we think that norms can be used to specify protocols, because they can dictate that in certain circumstances an agent ought to send a given type of message, or react to a message in a specific way, to comply with the regulations of a specific institution. How this can be done will be shown in Section 5.

We regard norms as event-driven rules that fire under appropriate conditions and, by doing so, create, update or cancel commitments affecting a predefined set of agents. At an abstract level, a norm is part of the definition of an artificial institution; its instances then regulate and are bounded to the organization that reifies the institution. Agents are liable to all the norms associated to the roles they play in an institution.

A norm is defined within an institution, observes an entity of an institution, is activated by an event concerning such an entity, and then fires if certain contextual conditions are met. Typically, interesting event types are the filling of a role by an agent, a value change of an institutional attribute, the reaching of certain instant of time, and so on.

When a norm fires, it is applied to a collection of *liable agents*, that are described by a suitable selection expression; in general, the collection of liable agents corresponds to the set of agents that play a given role in the institution. For every liable agent, the norm creates, updates or cancels a set of commitments.

The general structure of a norm can be described as follows:

within *context_name: identity*
on *e: event.type*

```

if contextual conditions then
  foreach agent in liable agent selection expression
  do commitmentActionDescription{;commitmentActionDescription}*

```

Many studies have been devoted to the analysis of the relationship holding between norms and commitments, which is often perceived as a fundamental aspect of institutions [10] and organizations [5]. For example in [11] commitments are viewed as a specialization of norms, while in [5] and [17] norms are a special kind of commitments, called *metacommitments*.

From our point of view, norms are not themselves commitments, but rules that manipulate commitments of the agents engaged in an interaction. In fact, norms are associated to roles rather than to individual agents; they do not have a debtor or a creditor, and strictly speaking they cannot be fulfilled or violated. Indeed, what can be fulfilled or violated is not a norm, but a commitment created by the application of a norm.

There are, in conclusion, two types of commitments: the ones created by individual agents through the execution of communicative acts, and the ones created by norms and acquired by an agent in virtue of its role in an institution.

5 The English Auction

In this section we will describe an example of a *special institution*, concerning the specification of a widely studied interaction framework: the English Auction. The formalization proposed exploits the conventional nature of this type of interaction, making explicit the social concepts and rules that constitute and regulate the interaction.

In the literature there are other attempts to specify the English Auction, like for instance the one proposed by FIPA [14] and the one presented in a previous work of ours [13]. But we think that the definition of the English Auction as a special institution overcomes some drawbacks of those formalizations. In particular in the approach presented in [14] commitments between the winner and the auctioneer are created only when the auction is closed. On the contrary in the current formalization and in [13] commitments are undertaken by the agents during the auction.

Another important advantage of this approach with respect to [13] is that the explicit formalization of the context of the interaction simplifies the content of the exchanged messages. For instance if the context is not made explicit, the auctioneer of an English Auction has to accept a bid of a participant, committing the auction house to give the product to that participant, on condition that no higher bids will be submitted. Otherwise the context can be made explicit for example by introducing the role *current_winner* and a norm that creates a commitment for the *current_winner* to pay the *ask_price* to the auction house, and a commitment for the auction house to give the *product* to the *current_winner* of the last round. Using this formalization, the exchanged messages for bidding and for declaring the winner are simple institutional actions, as will be shown in the next section.

5.1 The English Auction ontology (EAontology)

Entities The ontology of the English Auction consists of some institutional entities and is described by the class diagram reported in Figure 2. The fundamental entity called *EnglishAuction* is identified by its *id* and can assume three different states: *unset*, during the registration phase, *open* and *closed*. An *EnglishAuction* has a *product* that will be sold at the *ask_price*, which starts from the *reservation_price* and can only increase. During an English Auction there are periods of time, that we model by introducing the *Round* entity, during which agents can made their *Bids*. A *Round* has a fixed duration (*round_duration*) and it should be closed after its *close_time* has expired. If there are no bids in a *Round* or if the maximum number of rounds (*max_round*) is reached, the English Auction is closed.

An agent that takes part in an English Auction can fill the role of *participant*, or of *auctioneer*. Participants can only raise their *Bids* and the highest bidder is declared the *current_winner*. During one auction we assume that an agent cannot be both a participant and an auctioneer, while it must be a participant in order to be allowed to become a *current_winner*. Furthermore, all the agents that are related to the auction are gathered in the *EAGroup*.

Other concepts that are fundamental for every MAS, like *Agent* and *IndividualAgent*, are assumed to be defined in external ontologies.

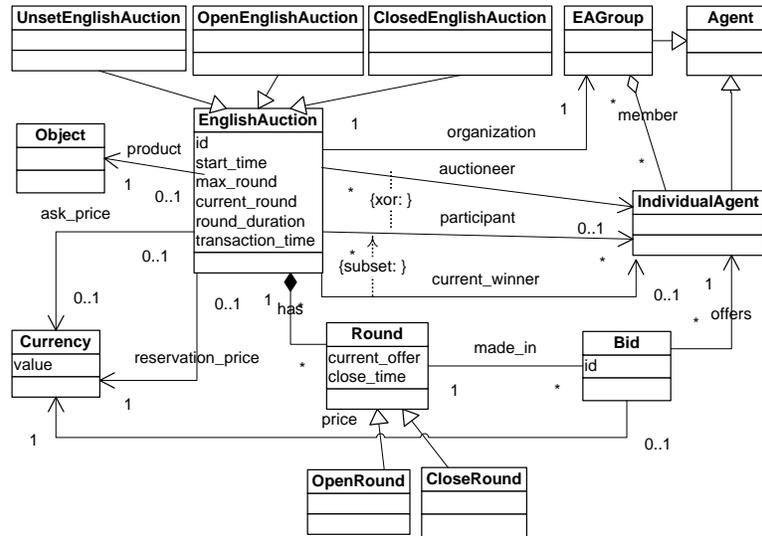


Fig. 2. Class diagram representing the English Auction ontology.

Institutional Actions The institutional actions that operate on *EnglishAuction* entities allow agents to open and close the auction, to make a bid, to set the *current_winner* or a new *ask_price*, to open and close a round.

To describe an institutional action a slight extension of OCL is needed. In fact, using the terminology introduced in [8], object oriented specifications usually treat actions as events, because they only model state changes in the world. Instead, an action is an event brought about by an agent, and may have different effects depending on which authorized agent has performed it. For example, the act of bidding creates a new offer for the bidder and not for other agents. Therefore, we introduce a new reserved word, *actor*, that is used to refer to the agent that is performing the action. Below we formally define the institutional actions made available by the EAontology. The action for opening an auction is:

```
name : openAuction(auct_id)
pre :   UnsetEnglishAuction.allInstances → exists(id = auct_id)
post :   OpenEnglishAuction.allInstances → exists(id = auct_id)
```

It can be successfully performed only if the *auct_id* corresponds to an auction not yet opened.

When an auction is opened, an agent can change the *ask_price*, that can only rise, executing the following action:

```
name : setAskPrice(auct_id, price)
pre :   OpenEnglishAuction.allInstances → exists(id = auct_id
and ask_price.value < price)
post :   OpenEnglishAuction.allInstances → exists(id = auct_id
and ask_price.value = price)
```

The phase when participants can place their bids is represented by the *Round* entity. An *EnglishAuction* can have only one *Round* open. In our formalization, opening a bid session correspond to create a new *Round*, calculate when it should be closed and increase the *current_round* counter.

```
name : newRound(auct_id)
pre :   not OpenEnglishAuction.allInstances → select(id = auct_id)
.round → exists(oclIsTypeOf(OpenRound))
post :   let a : EnglishAuction = EnglishAuction.allInstances → select(
id = auct_id)
a.round → select(r | not a.round@pre → including(r) and
r.oclIsTypeOf(OpenRound) and close_time = now + a.round_duration))
→ sizeOf() = 1 and a.current_round = a.current_round@pre + 1
```

During a round, an agent can only make one bid with a price higher than the *ask_price*. A successful act of bidding increments the number of offers ,that is used to identify a new *Bid* of the actor with the offered price.

```

name : makeBid(auct_id, price)
pre : let a: OpenEnglishAuction = OpenEnglishAuction.allInstances →
      select(id = auct_id)
      a.round → select(r | r.oclIsTypeOf(OpenRound)).bid →
      select(actor = offers) → isEmpty() and a.ask_price.value < price)
post : OpenEnglishAuction.allInstances → select(id = auct_id).round
      → select(r.oclIsTypeOf(OpenRound) and r.current_offer
      = round.current_offer@pre + 1 and r.bid → select(b |
      not r.bid@pre → including(b) and b.offers = actor and
      b.price.value = price) → sizeOf() = 1))

```

A participant can be declared the *current_winner* only if he has made the highest bid.

```

name : currentWinnerIs(auct_id, agent)
pre : let a:OpenEnglishAuction = OpenEnglishAuction.allInstances
      → select(id = auct_id)
      a.round.bid → select(b | b.price.value = a.round.bid.price.value → max())
      .offers = agent
post : OpenEnglishAuction.allInstances → select(id = auct_id)
      .current_winner = agent

```

An agent can close an open round, and, as a consequence of this action, all rounds associated with the auction result closed:

```

name : closeRound(auct_id)
pre : OpenEnglishAuction.allInstances → select(id = auct_id).round
      → exists(oclIsTypeOf(OpenRound))
post : OpenEnglishAuction.allInstances → select(id = auct_id).round
      → forAll(oclIsTypeOf(ClosedRound))

```

Finally, an agent can close an open auction:

```

name : closeAuction(auct_id)
pre : OpenEnglishAuction.allInstances → exists(id = auct_id)
post : CloseEnglishAuction.allInstances → exists(id = auct_id)

```

5.2 Authorizations

A participant is authorized only to make bids:

```
Auth(EnglishAuctionid.participant, makeBid(id, price))
```

Auctioneers are authorized to perform all other actions defined by the EAOntology. Some of these authorizations are conditioned, for example an auctioneer is authorized to open an auction only if its *start_time* has elapsed and if there are at least two agents registered as participants:

$\text{Auth}(\text{EnglishAuction}_{id}.\text{auctioneer}, \text{openAuction}(id),$
 $\text{now} > \text{EnglishAuction}_{id}.\text{start_time} \text{ and } \text{EnglishAuction}_{id}.\text{participant} \rightarrow \text{size}() > 2)$

In a similar way, an auctioneer is authorized to close a round only if its *close_time* has elapsed.

$\text{Auth}(\text{EnglishAuction}_{id}.\text{auctioneer}, \text{closeRound}(id),$
 $\text{now} > \text{EnglishAuction}_{id}.\text{round} \rightarrow \text{select}(r.\text{oclIsTypeOf}(\text{OpenRound})).\text{close_time})$

The auctioneer is authorized to perform the remaining actions without any constraint.

$\text{Auth}(\text{EnglishAuction}_{id}.\text{auctioneer}, \text{setAskPrice}(id, \text{price}))$
 $\text{Auth}(\text{EnglishAuction}_{id}.\text{auctioneer}, \text{closeAuction}(id))$
 $\text{Auth}(\text{EnglishAuction}_{id}.\text{auctioneer}, \text{newRound}(id))$
 $\text{Auth}(\text{EnglishAuction}_{id}.\text{auctioneer}, \text{currentWinners}(id, \text{agent}))$

5.3 Event types

Before describing norms in details, we need to discuss further how to model event types and event tokens, an individual belonging to an event type (see [8]), in order to describe more precisely which kind of events activates a norm and how norms can detect their occurrences.

The Unified Modeling Language (UML) [4] models four kinds of events: *signals*, *calls*, *passing of time* and *change in state*. Unfortunately, the notation proposed in UML for modeling events is bound to features of State Machine and Statechart Diagrams, while we need a way to describe events in general. Inspired by UML notation for signals, here we propose to model type of events as stereotyped classes. A stereotype extends the UML vocabulary, creating new building blocks that are derived from existing ones. In this case, *event* stereotype is derived by UML *class*, that we use also to represent physical objects and institutional entity. As classes, an event type can have attributes, providing information about the state transition that caused it, and it is possible to model hierarchies of events. Furthermore, we assume that the system returns the time at which an event token has occurred, by means of the *time_of()* operator.

In our formalization we have singled out three three main categories of events:

- *TimeEvent*, that occurs when the system reaches a certain *instant_of_time*.
- *ChangeEvent*, that happens when an institutional entity changes in some way. This kind of event type can be specialized further:
 - *InstitutionalPropertyChange* is registered when an attribute has changed its value.
 - *InstitutionalRelationChange*, that is, when a new relation is *created* or an existing one between the institutional entity and another one is *dropped*.
 - *InstitutionalStateChange* occurs when an entity modifies its type in a given taxonomy. For example, when an auction from *unset* becomes *open*.

- *ActionEvent* happens when an agent perform an action. In particular, an interesting type of this kind of events is *ExchangeMessage*, that represents the act of sending a message.

These main categories and their specializations are represented in Figure 3.

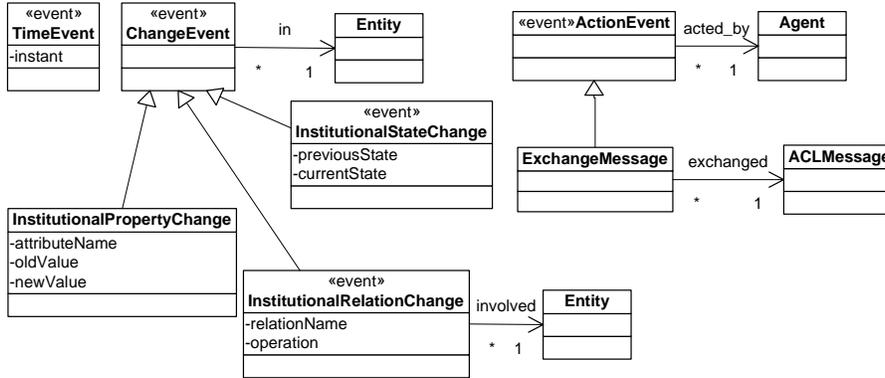


Fig. 3. Class diagram representing main event types.

The definition of event types allow us to describe event tokens and *event templates*, that are, event types with some restriction on certain attributes. The main difference between event token and event template is that the first describes only one occurrence, while the latter describes a set of possible event occurrences.

Event descriptor are used in the *on* section of a norm: when an event matches the given descriptor, the corresponding norm is fired and its variable *e* is filled with the event that activated it.

5.4 Norms

In this section, we will describe the set of norms that dictate the behaviour of agents that has joined an interaction system regulated by the English Auction Institution. Each norm prescribes what agents should or should not do in correspondence of relevant institutional events. To prevent system overload due to the exchange of useless messages, we will assume that exists a special norm that forbids registered agents to execute those institutional actions that are ontologically impossible or that are not authorized. We will not present such norm explicitly, because its formalization requires some concepts related to commitment content that has not discussed in this paper.

If this set of norms is respected, the consequent interaction can be described with the interaction diagram of Figure 4, where in each state there is represented the value of main attributes and edges indicate the successful performance of an institutional action by an authorized actor.

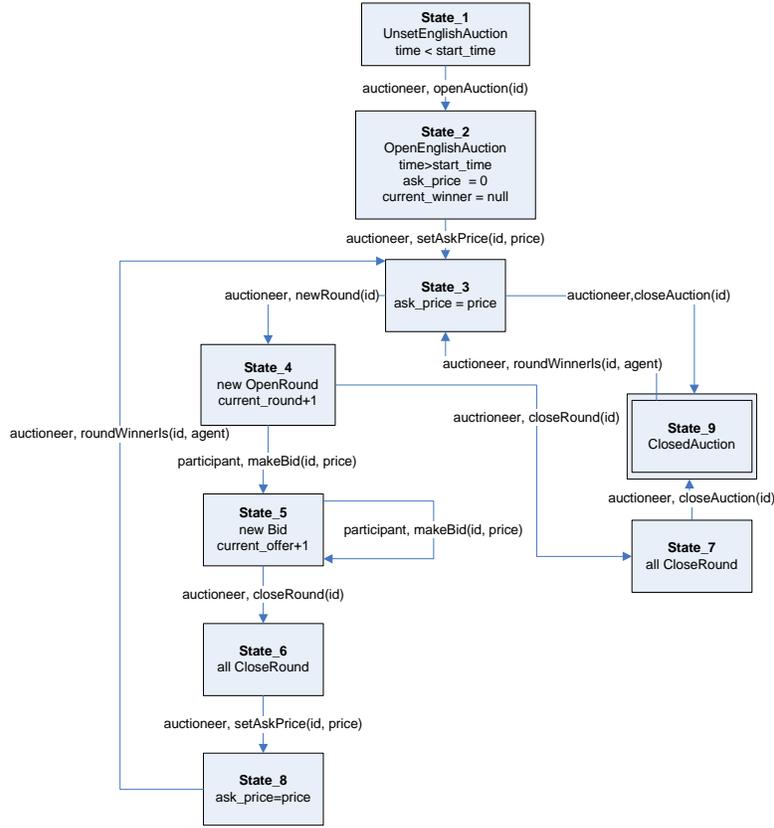


Fig. 4. The protocol prescribed by norms of the English Auction Institution

State_1 corresponds to the registration phase, during which agents assume the role of participants or auctioneer. When the *start_time* is elapsed and if two agents has been registered as participants, a norm create an obligation for the auctioneer to open the auction:

```

within a: UnsetEnglishAuction
on e: TimeEvent(a.startTime)
if a.participant.sizeOf() >= 2 then
  foreach agent in a.auctioneer
  do
    makePendingComm(agent, a.organization,
      (openAuction(a.id), [now, now +  $\delta$ ],  $\exists$ ))
  
```

Where δ is the time allowed to the auctioneer to fulfill its obligation.

It is important to remark that norms concerning the auctioneer have a recurrent pattern. In fact, at every stage of the interaction, the auctioneer is obliged

to do a specific action among those that are ontologically possible and it is authorized to perform, whereas it is forbidden from doing any of the others. This pattern can be recognized in the following norm:

```

within a: EnglishAuction
on e: InstitutionalStateChange(a, UnsetAuction, OpenAuction)
if a.ask_price = 0 then
  foreach agent in a.auctioneer
  do
    makePendingComm(agent, a.organization,
      (setAskPrice(a.id, a.reservation_price), [now, now +  $\delta$ ],  $\exists$ ))
    makePendingComm(agent, a.organization,
      (not newRound(a.id), [now, time_of(e :
        InstitutionalStateChange(a, OpenAuction, ClosedAuction))],  $\forall$ ))

```

The previous norm states that when the auction is declared open (*State_2*), the auctioneer should set the reservation price. Furthermore, it forbids the auctioneer from opening a round until the auction is closed. Actually, that prohibition is cancelled by another norm when the auctioneer sets a new *ask_price* and an obligation to open a round of offers is created:

```

within a: EnglishAuction
on e: InstitutionalRelationChange(a.ask_price, created)
if a.current_round < max_round then
  foreach agent in a.auctioneer
  do
    foreach comm in Comm.AllInstances  $\rightarrow$  select(debtor = agent and
      creditor = a.organization and
      content.match(not newRound(a.id)))
    do
      setCancel(comm.debtor, comm.creditor, comm.content)
      makePendingComm(agent, a.organization,
        (newRound(a.id), [now, now +  $\delta$ ],  $\exists$ ))
      makePendingComm(agent, r.englishAuction.organization,
        (not setAskPrice(a.id, -), [now, time_of(e :
          InstitutionalStateChange(a, OpenAuction, ClosedAuction))],  $\forall$ ))
      makePendingComm(agent, r.englishAuction.organization,
        (not closeRound(a.id), [now, time_of(e :
          InstitutionalStateChange(a, OpenAuction, ClosedAuction))],  $\forall$ ))
      makePendingComm(agent, r.englishAuction.organization,
        (not currentWinnerIs(a.id, -), [now, time_of(e :
          InstitutionalStateChange(a, OpenAuction, ClosedAuction))],  $\forall$ ))

```

When the auctioneer declare a new *ask_price* (*State_3*), a set of authorized and ontologically possible actions should be prohibited, because we assume that

the *current_winner* and the *ask_price* should be declared only when a Round is closed. Instead, an auction can be closed only when the *current_round* has reach the *max_round* or when it has not received any bid from participants as described by the following two norms:

```

within a: EnglishAuction
on e: InstitutionalRelationChange(a.ask_price, created)
if a.current_round = max_round then
  foreach agent in a.auctioneer
  do
    foreach comm in Comm.AllInstances  $\rightarrow$  select(debtor = agent and content.match(not
      closeAuction(r.englishAuction.id))
    do
      setCancel(comm.debtor, comm.creditor, comm.content)
      makePendingComm(agent, a.organization,
        (closeAuction(a.id), [now, now +  $\delta$ ],  $\exists$ ))

```

```

within r: Round
on e: InstitutionalStateChange(r, OpenRound, CloseRound)
if r.bid  $\rightarrow$  isEmpty() then
  foreach agent in r.englishAuction.auctioneer
  do
    foreach comm in Comm.AllInstances  $\rightarrow$  select(debtor = agent and content.match(not
      closeAuction(r.englishAuction.id))
    do
      setCancel(comm.debtor, comm.creditor, comm.content)
      makePendingComm(agent, r.englishAuction.organization,
        (closeAuction(r.englishAuction.id), [now, now +  $\delta$ ],  $\exists$ ))

```

When a new round is opened (*State₄*), preconditions of *makeBid* become true and participants can make only one bid performing that institutional action . Unlike the auctioneer, a participant has the permission to make a bid, but it is not obliged.

After the *round_duration* has elapsed, the auctioneer is committed to close the round:

```

within r: Round
on e: TimeEvent(r.close_time)
if r.oclIsTypeOf(OpenRound) then
  foreach agent in a.auctioneer
  do
    foreach comm in Comm.AllInstances  $\rightarrow$  select(debtor = agent and
      creditor = r.englishAuction.organization and
      content.match(not closeRound(r.englishAuction.id)))
    do

```

```

    setCancel(comm.debtor, comm.creditor, comm.content)
    makePendingComm(agent, a.organization,
        (closeRound(a.id), [now, now +  $\delta$ ],  $\exists$ ))

```

If there is a valid offer, the auctioneer is obliged to proclaim *current_winner* the agent that has made the highest bid and to set the value offered as the new *ask_price* (*State_6* and *State_8*).

```

within a: EnglishAuction
on e: InstitutionalRelationChange(a.ask_price, created)
if a.round  $\rightarrow$  notEmpty() then
    foreach agent in a.auctioneer
    do
        makePendingComm(agent, a.organization,
            (currentWinnerIs(a.round.bid  $\rightarrow$  select
                ((b1, b2 | b1 <> b2 implies
                    ((b1.price.value > b2.price.value) or (b1.price.value =
                    b2.price.value and b1.id > b2.id).offers), [now, now +  $\delta$ ],  $\exists$ )))

```

```

within r: Round
on e: InstitutionalStateChange(r, OpenRound, CloseRound)
if not r.bid  $\rightarrow$  isEmpty() then
    foreach agent in r.englishAuction.auctioneer
    do
        foreach comm in Comm.AllInstances  $\rightarrow$  select(debtor = agent and
            content.match(notsetAskPrice(r.englishAuction, -))
        do
            setCancel(comm.debtor, comm.creditor, comm.content)
            makePendingComm(agent, r.englishAuction.organization,
                (setAskPrice(r.englishAuction, r.bid.price.value  $\rightarrow$  max()),
                , [now, now +  $\delta$ ],  $\exists$ ))

```

Every time an agent is declared *current_winner*, it is committed to buy the *product* on sale at the price of its last bid, whereas a similar commitment with debtor the previous winner is cancelled.

```

within a: EnglishAuction
on e: InstitutionalRelationChange(a, current_winner, created)
if true then
    foreach agent in a.current_winner
    do
        makePendingComm(agent, a.auctioneer,
            (give(agent, a.auctioneer, a.ask_price),
            [time_of(e : InstitutionalStateChange(a, OpenAuction, ClosedAuction)),

```

time_of($e : \text{InstitutionalStateChange}(a, \text{OpenAuction}, \text{ClosedAuction})$)
 $+a.\text{transactionTime}$], \exists)

```

within  $a$ : EnglishAuction
on  $e$ : InstitutionalRelationChange( $a$ ,  $\text{current\_winner}$ ,  $\text{create}$ )
if true then
  foreach  $agent$  in  $a.\text{participant}$ 
  do
    foreach  $comm$  in  $\text{Comm.AllInstances} \rightarrow \text{select}(\text{debtor} = agent \text{ and}$ 
       $\text{creditor} = a.\text{auctioneer} \text{ and}$ 
       $\text{content.match}(\text{give}(agent, a.\text{auctioneer}, -))$ )
    do
       $\text{setCancel}(comm.\text{debtor}, comm.\text{creditor}, comm.\text{content})$ 

```

At the same moment, an obligation for the auctioneer to give the *product* to the previous *current_winner* is canceled, whereas a similar one is created with the new *current_winner* as the creditor.

```

within  $a$ : EnglishAuction
on  $e$ : InstitutionalRelationChange( $a$ ,  $\text{current\_winner}$ ,  $\text{create}$ )
if true then
  foreach  $agent$  in  $a.\text{auctioneer}$ 
  do
    foreach  $comm$  in  $\text{Comm.AllInstances} \rightarrow \text{select}(\text{debtor} = agent \text{ and}$ 
       $\text{content.match}(\text{give}(agent, -, a.\text{product}))$ )
    do
       $\text{setCancel}(comm.\text{debtor}, comm.\text{creditor}, comm.\text{content})$ 
       $\text{makePendingComm}(agent, a.\text{current\_winner},$ 
         $(\text{give}(agent, a.\text{current\_winner}, a.\text{product}), [\text{time\_of}(e :$ 
           $\text{InstitutionalStateChange}(a, \text{OpenAuction}, \text{ClosedAuction})$ ],  $\text{time\_of}(e :$ 
           $\text{InstitutionalStateChange}(a, \text{OpenAuction}, \text{ClosedAuction})$ )] +
           $a.\text{transaction\_time}$ ],  $\exists$ )
       $\text{makePendingComm}(agent, a.\text{organization},$ 
         $(\text{not } \text{currentWinnerIs}(a.\text{id}, -), [\text{now}, \text{time\_of}(e :$ 
           $\text{InstitutionalStateChange}(a, \text{OpenAuction}, \text{ClosedAuction})$ ]]),  $\forall$ )

```

6 Conclusions

In this paper we have defined an ACL as a set of conventions to act on a fragment of institutional reality, defined in the context of an artificial institution, called the Basic Institution. Within such an approach, we proposed a commitment-based semantics for an ACL, and showed that all commonly used types of communicative acts can be defined in terms of a single basic type, namely declarations.

Then we have defined special institutions, that augment the Basic Institution by adding ontological and normative elements, and showed how a well known interaction framework, the English Auction, can be regarded as a special institution, its interaction protocol being defined as a set of norms.

We believe that our approach helps clarifying the strict relationships holding between language, institutional reality, and interaction rules in a MAS. Moreover, we believe that the adoption of an operational modelling style makes our proposal reasonably easy to implement. In fact, we plan to implement our framework as an extension of JADE in the near future.

References

1. L. Amgoud, N. Maudet, and S. Parsons. An argumentation-based semantics for agent communication languages. In F. V. Harmelen, editor, *Proceedings of the European Conference on Artificial Intelligence (ECAI-2002)*, pages 38–42, Lyon, France, July 2002. IOS Press.
2. M. Barbuceanu, T. Gray, and S. Mankovski. Coordinating with obligations. In K. P. Sycara and M. Wooldridge, editors, *Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98)*, pages 62–69, New York, 1998. ACM Press.
3. J. Bentahar, B. Moulin, and B. Chaib-draa. Commitment and argument network: A new formalism for agent communication. In F. Dignum, editor, *Advances in Agent Communication, International Workshop on Agent Communication Languages, ACL 2003, Melbourne, 2003*, volume 2922 of *LNCS*, pages 146–165. Springer, 2004.
4. G. Booch, J. Rumbaugh, and I. Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley, Reading, Massachusetts, USA, 1 edition, 1999.
5. C. Castelfranchi. Commitments: From individual intentions to groups and organizations. In V. Lesser, editor, *Proc. First International Conference on Multi-Agent Systems*, pages 528–535, San Francisco, USA, 1995. AAAI-Press and MIT Press.
6. M. Colombetti. 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)*, pages 21–29, Barcelona, Spain, 2000.
7. M. Colombetti, N. Fornara, and M. Verdicchio. A social approach to communication in multiagent systems. In J. A. Leite, A. Omicini, L. Sterling, and P. Torroni, editors, *Declarative Agent Languages and Technologies*, volume 2990 of *LNAI*, pages 121–150, to be published. Springer, 2004.
8. M. Colombetti and M. Verdicchio. An analysis of agent speech acts as institutional actions. In C. Castelfranchi and W. L. Johnson, editors, *Proc. First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2002)*, pages 1157–1166, Bologna, Italy, 2002. ACM Press.
9. F. Dignum and L. Royakkers. Collective obligation and commitment. In *In Proc. of 5th Int. conference on Law in the Information Society*, Florence, Italy, 1998.
10. M. Esteva, J. A. Rodríguez-Aguilar, C. Sierra, P. Garcia, and J. L. Arcos. On the formal specification of electronic institutions. In F. Dignum and C. Sierra, editors, *Agent Mediated Electronic Commerce, The European AgentLink Perspective (LNAI 1991)*, pages 126–147. Springer, 2001.

11. F. Lopez y Lopez and M. Luck. Modelling Norms for Autonomous Agents. In E. Chavez, J. Favela, M. Mejia, and A. Oliart, editors, *Proceedings of Fourth Mexican International Conference on Computer Science*, pages 238–245, 2003.
12. N. Fornara and M. Colombetti. Operational specification of a commitment-based agent communication language. In C. Castelfranchi and W. L. Johnson, editors, *Proc. First International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2002)*, pages 535–542, Bologna, Italy, 2002. ACM Press.
13. N. Fornara and M. Colombetti. Defining interaction protocols using a commitment-based agent communication language. In J. S. Rosenschein, T. Sandholm, M. Wooldridge, and M. Yokoo, editors, *Proc. Second International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2003)*, pages 520–527, Melbourne, Australia, 2003. ACM Press.
14. Foundation for Intelligent Physical Agents. FIPA English Auction Interaction Protocol Specification. <http://www.fipa.org>, 2001.
15. Foundation for Intelligent Physical Agents. FIPA Communicative Act Library Specification. <http://www.fipa.org>, 2002.
16. A. Jones and M. J. Sergot. A formal characterisation of institutionalised power. *Journal of the IGPL*, 4(3):429–445, 1996.
17. M. P. Singh. An ontology for commitments in multiagent systems: Toward a unification of normative concepts. *Artificial Intelligence and Law*, 7:97–113, 1999.
18. P. McBurney and S. Parsons. Posit spaces: a performative model of e-commerce. In J. S. Rosenschein, T. Sandholm, M. Wooldridge, and M. Yokoo, editors, *Proc. Second International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2003)*, pages 624–631. ACM Press, 2003.
19. Y. Moses and M. Tennenholtz. Artificial social systems. *Computers and AI*, 14(6):533–562, 1995.
20. Object Management Group, OMG . Object Constraint Language Specification 1.4. <http://www.omg.org/>, 2003.
21. J. R. Searle. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, Cambridge, United Kingdom, 1969.
22. J. R. Searle. *The construction of social reality*. Free Press, New York, 1995.
23. J. R. Searle and D. Vanderveken. *Foundations of Illocutionary Logic*. Cambridge University Press, Cambridge, UK, 1984.
24. M. P. Singh. A social semantics for agent communication languages. In *Proceedings of IJCAI-99 Workshop on Agent Communication Languages*, pages 75–88, 1999.
25. J. M. V. Marik and M. Pechoucek, editors. *Modelling Electronic Organizations*, volume 2691 of *LNAI*. Springer, 2003.
26. M. Verdicchio and M. Colombetti. A logical model of social commitment for agent communication. In J. S. Rosenschein, T. Sandholm, M. Wooldridge, and M. Yokoo, editors, *Proc. Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2003)*, pages 528–535, Melbourne, Australia, 2003. ACM Press.

Technical report No. 1

Agent Communication and Institutional Reality

N. Fornara, F. Viganò, M. Colombetti

2004

ITC

**Institute for
Communication
Technologies**

www.itc.com.unisi.ch

Faculty of
Communication
Sciences

Università della
Svizzera
italiana

via Buffi 13
CH-6900
Lugano