Agent-Mediated Interaction. From Auctions to Negotiation and Argumentation

Carles Sierra, Pablo Noriega

Artificial Intelligence Research Institute —IIIA,
Spanish Council for Scientific Research —CSIC,
08193 Bellaterra, Barcelona, Catalonia, Spain.
sierra@iiia.csic.es, pablo@iiia.csic.es

Abstract. Most approaches to modelling agent interactions tend to focus just on the mechanism: the protocol and language used for the interaction, and forget the context where that interaction takes place. We hold that although the complexity of the problem to be solved is associated to the complexity of the mechanism, modelling the mechanism alone is insufficient. We argue that an appropriate representation of the context and pragmatics associated to the interaction, as well as a practical way of enforcing the accepted interaction conventions are essential for the design of successful MAS applications. The concept of Electronic Institution is presented both as a way to reconcile mechanisms with their corresponding pragmatic and contextual aspects, and a way of extending familiar notions of mediation to MAS.

1 Introduction

In this paper we explore two dimensions that we believe should be present in the design of multi-agent systems. One is the relationship between the complexity of the problem that participating agents attempt to solve, and the complexity of the interactions among them; more specifically, the complexity of the language these agents use to communicate. The other dimension has to do with the use of some sort of mediation among participating agents in order to achieve, or facilitate, successful interactions. By exploring these two dimensions in three familiar examples of agent interactions –auctions, structured negotiation and persuasive argumentation– we will argue in favour of the notion of an Electronic Institution as a powerful device to handle complex agent interactions. We will show that electronic institutions are a natural device to make explicit not only the interaction mechanisms required in MAS, but also the contextual and pragmatic features that are needed for adequate MAS modelling of complex problem solving tasks.

2 Interaction Mechanisms. The myopic view

In this section three increasingly complex interaction mechanisms are briefly overviewed. We will see how the increasing complexity of the mechanisms leads to the solution of more complex problems, and we will argue that ignoring the pragmatics of the interaction makes the mechanisms almost inapplicable in real settings.

2.1 Auctions

When participating in auctions, buying agents use a rather simple language to communicate, both in terms of the illocutionary particles they hear or utter — basically *offer* goods and *accept* prices— and of the content language: in most auctions, the buyer simply accepts a posted price, the auctioneer, hence, needs to communicate buyers only those elements that characterize the item that is being auctioned and the current bidding-price. Overall an extremely simple ontology. In fact, quite similar to the language of ants in an ant algorithm! The problem to be solved is the allocation of the good to the buyer that values the good most, or more precisely, to the buyer that is ready to pay more for the good. A simple language for an, in principle, simple *unidimensional search problem*.

To more accurately illustrate this simple communication language here is a brief description of a real auction house.

A Fish Auction House in the Catalonian Seashore Twice a day the fishing fleet of the town of Blanes (Costa Brava) sells its catch in the local market place. The market is managed by the fishermen's guild under a lease from the government. Fishermen get their revenues from the auctioning of their catches using the traditional *Dutch*[1] auction protocol (see Figure 1). The Blanes market is nowadays mechanised to a certain extent: it has a panel showing the high pace decreasing prices of the auctioned item, and information about the boat and the boxes being auctioned; infra-red sensors permit buyers wandering around to stop the decreasing pace by pointing with a personal infra-red bidder to the ceiling of the room; a complete information system to support registering, payment, credits and so on, is maintained by the auction house. However, although the Blanes fish market has evolved to a certain degree, its main features are the traditional ones (documents on this specific market date back to the XVth Century). For a complete account of the details of this example of an auction house refer to [13].

There are three types of participants in this auction house: *sellers*, *buyers* and different types of *market intermediaries*. They interact in order to perform a series of activities:

- **Sellers and buyers admission** This activity is performed by a market intermediary (that is usually different for buyers and sellers) that authenticates, manages the credit of buyers and sellers, and is responsible for the

[1] This downward bidding protocol is called *Dutch* because it is similar to the one used in Holland to sell flowers.

assignment of bidding devices to the buyers. The sellers' admitter is responsible for the registration process of the goods on sale (assignment of initial auctioning prices and quality of fish) and on the sequencing of the boxes to be auctioned.

- **Sellers and buyers settlements** This activity is mediated by another type of staff members who are responsible for invoicing and payment procedures to and from sellers and buyers.
- **Auctioning** Again, a scene mediated by a staff auctioneer. This activity involves the actual assignment of boxes to buyers according to the Dutch protocol. The auctioneer takes care of incidents like collisions (more than one buyer bid at the same price), invalid bids (someone made a bid before the auction clock was started), or cancellation due to unforeseen circumstances (malfunctioning of a device, erroneous bid, ...).

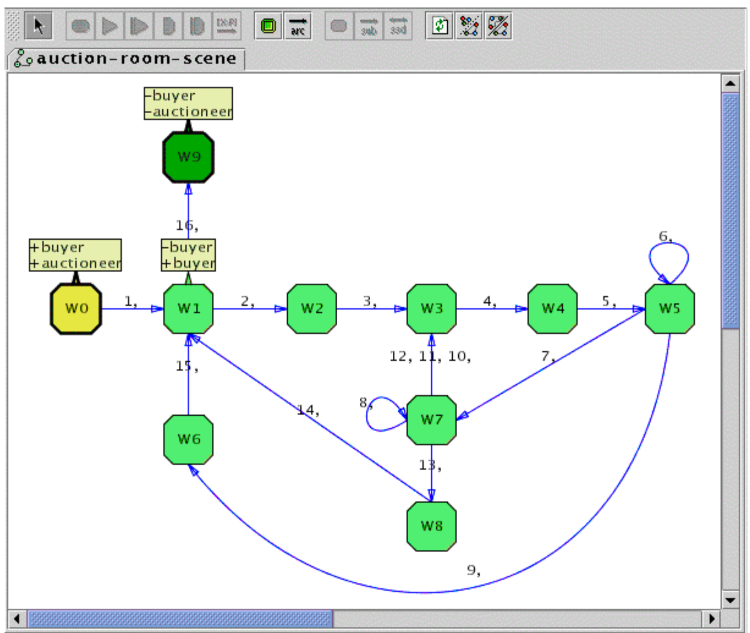

Fig. 1. Dutch Auction Bidding Protocol Diagram. It involves the roles of an *auctioneer* and *buyer* agents. Numbers denote illocution schemata as explained in the text of the paper; for instance, label 1 corresponds to $inform(?x : auctioneer, all : buyer, open\_auction(?n))$. As in other figures, boxes represent conversation states, arrows represent transitions triggered by the utterance of the illocutions matching the corresponding label, and rectangles denote access (+) and exit (-) states for participating roles.

These activities are dialogical in the sense that all observable interactions are tagged with an illocution, an interaction has social consequences only if tagged by an illocution, and only admissible illocutions are uttered and uttered according to explicit conventions. Notice that the type of electronic auctioning used in Blanes could be argued not to be dialogical in *stricto-senso* since there is an electronic device mediating between the buyer and the auctioneer, although its role is a mere substitution of the old time dialogical utterances by equivalent infra-red emissions.

As mentioned before, if we look in detail into the auctioning activity —or, more appropriately, the bidding *scene*[2]— we see that the language needed to model it is rather simple: the auctioneer (either verbally or through the panel) announces a good to be auctioned, which can be modelled as a term in a simple fish ontology, and then there is a sequence of prices from the auctioneer to buyers and a signal (the actual *mine* uttered in a traditional live auction) from a buyer to the auctioneer. Nothing too complex.

The language used by the interacting agents is simple because the problem to solve is simple, but *also* for empirical reasons:

- Fish must be sold quickly. The use of protocols not bounded in time, like for instance an English protocol, would be impractical.
- Only price is involved. The remaining elements of the transaction: weight of the box and quality, are fixed by the auction house.

Here is the actual list of illocutions that model the bidding rounds in a Dutch auction, as labelled in Figure 1:

$$\begin{array}{rc}
1 & inform(?x : auctioneer, all : buyer, open_auction(?n)) \\
2 & inform(!x, all : buyer, open_round(?r)) \\
3 & inform(!x, all : buyer, to_sell(?good_id)) \\
4 & inform(!x, all : buyer, buyers(?b)) \\
5 & inform(!x, all : buyer, offer(!good_id, ?price)) \\
6 & inform(!x, all : buyer, offer(!good_id, ?price)) \\
7 & commit(?y : buyer, !x : auctioneer, bid(!good_id, ?price)) \\
8 & commit(?y : buyer, !x : auctioneer, bid(!good_id, ?price)) \\
9 & inform(!x, all : buyer, withdrawn(!good_id, ?price)) \\
10 & inform(!x, all : buyer, collision(?price)) \\
11 & inform(!x, all : buyer, sanction(?buyer_id)) \\
12 & inform(!x, all : buyer, expulsion(?buyer_id)) \\
13 & inform(!x, all : buyer, sold(!good_id, ?price, ?buyer_id))
\end{array}$$

[2] We'll refer to these dialogical activities as *scenes* because they are performed or *enacted* much like scenes in the script of a theater play are performed: agents (actors) engage in dialogues according to their pre-assigned role (character).

14	$inform(!x, all : buyer, end\_round(?r))$
15	$inform(!x, all : buyer, end\_round(!r))$
16	$inform(!x, all : buyer, end\_auction(!n))$

It should be noted that although this list of illocutions –and their pragmatic impact– may adequately model the auction scene, it is by no means sufficient to properly model a complete *auction house*, as we shall show below.

2.2 Negotiation

Other negotiation protocols are used to solve more complex problems: service agreements, selection of products and their associated payment conditions, or to agree on a co-ordinated action to perform a task. Such protocols are defined in order to facilitate agents the exploration of the space of potential agreements. Mathematically speaking, this means a set of points in a multidimensional space defined by the attributes of the negotiation object, such as, for instance, conditions of the service, characteristics of products, or responsible agents for given actions. Auctions are, in a sense, degenerated negotiation dialogues where the only negotiable dimension is price. This is the reason why dialogues in auctions tend to be rather simple. To illustrate a slightly more complex type of negotiation, specially in terms of content of the illocutions, see in Figure 2 a standard negotiation protocol –the one we propose for supply chain negotiations– and compare it with the auction protocol in Figure 1.

The fact that we move from a one-dimensional space into a multi-dimensional one makes the agent interaction language change substantially. The dialogue, although pragmatically remains almost identical –*offers* and *acceptance* of offers– is now open to new strategic possibilities like making trade-offs between attributes, deciding to concede less on important attributes and more on the less important ones, using time as a relevant element in defending a deal, ... The language gets more complex for a more complex problem: *distributed search for an agreement in multidimensional spaces.*

While in auctions it can be argued that the proper strategy of a buyer is to simply bid the perceived value of an item, the strategic decisions are far more involved for participants in the type of structured negotiation we are mentioning[23]. Negotiating agents need to decide not only if an offer is acceptable, but what counteroffer to respond with, and what criteria are more relevant to consider in the computation of counteroffers in a given environment: remaining negotiation time?, available resources? Also, the guessing of the opponent preferences (or the assumption of some default preferences as explored in [5]), or past experiences, may be of critical importance in order to develop successful negotiation strategies.

Summarizing, in terms of the problem to be solved negotiation can be viewed as a distributed search through a space of potential agreements [7,10]. The dimensionality and topology of this space is determined by the structure of the negotiation object. In terms of language complexity we observe an increase in the complexity of the illocution content although the pragmatics are not specially more sophisticated than in the auction case.

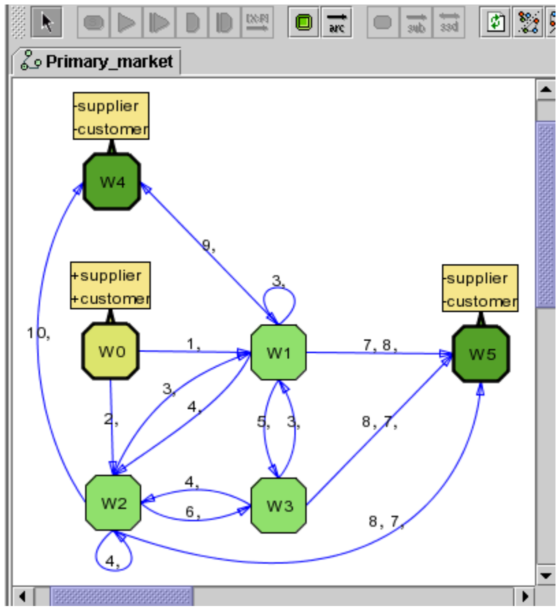

Fig. 2. Negotiation Protocol diagram.

Supply chain negotiations A common example for negotiation interaction between agents is that of Supply Chains. Supply chains have been a traditional focus of attention in the design of multi-agent systems [21] because of the significant role supply chains play in the structuring of the manufacturing economy and because they are a naturally distributed system where agents try to maximise their own profit, therefore permitting a classical economical analysis of their strategies. Thus, a MAS in supply chain will typically consist of a group of selfish agents that will trade by buying one level below in the chain, adding value to the purchased goods, and selling the improved good up to the next level in the chain. Although the true model is generally a supply tree or a supply graph, a simple supply chain is rich enough to show the potential complexities of a MAS design process.

Let us consider an example of a supply chain consisting of three levels: S_1, processing rough materials to produce goods to be sold to level S_2 which, in turn, processes the goods bought to S_1 to re-sell them to the final consumers represented as level S_3. The mechanism that the agents use to buy and sell products is a one-to-one negotiation. The agent model must specify a range of strategies and tactics that agents can employ to generate initial offers, evaluate proposals and offer counter proposals. Figure 2 shows a protocol that could be used to model this sort of negotiation, where the buyers play the *customer* role and sellers play the role of *supplier*. The following list contains the illocutions associated to the labels in figure 2:

1 $\mathit{offer}(?s : \mathit{supplier}, ?c : \mathit{customer}, \mathit{sign}(?d : \mathit{deal}, ?\mathit{date}), ?\mathit{date})$
2 $\mathit{offer}(?c : \mathit{customer}, ?s : \mathit{supplier}, \mathit{sign}(?d : \mathit{deal}, ?\mathit{date}), ?\mathit{date})$
3 $\mathit{offer}(!s : \mathit{supplier}, !c : \mathit{customer}, \mathit{sign}(?d : \mathit{deal}, ?\mathit{date}), ?\mathit{date})$
4 $\mathit{offer}(!c : \mathit{customer}, !s : \mathit{supplier}, \mathit{sign}(?d : \mathit{deal}, ?\mathit{date}), ?\mathit{date})$
5 $\mathit{reject}(!c\mathit{customer}, !s : \mathit{supplier}, \mathit{sign}(!d : \mathit{deal}, !\mathit{date}))$
6 $\mathit{reject}(!s : \mathit{supplier}, !c\mathit{customer}, \mathit{sign}(!d : \mathit{deal}, !\mathit{date}))$
7 $\mathit{withdraw}(!c : \mathit{customer}, !s : \mathit{supplier})$
8 $\mathit{withdraw}(!s : \mathit{supplier}, !c : \mathit{customer})$
9 $\mathit{accept}(!c : \mathit{customer}, !s : \mathit{supplier}, \mathit{sign}(!d : \mathit{deal}, !\mathit{date}))$
10 $\mathit{accept}(!s : \mathit{supplier}, !c : \mathit{customer}, \mathit{sign}(!d : \mathit{deal}, !\mathit{date}))$

2.3 Argumentation

Argumentation is a key form of interaction in multi-agent systems where independent agents behave autonomously. Since under those conditions agents have no direct control on one another, the only way they can influence one another's behaviour is through persuasion. In this situation, agents are not, as in negotiation, looking for a point in a multidimensional space that is acceptable to both parties, but rather trying to change the opponent's mind in order to change his/her preferences, beliefs or goals. The pragmatics of the dialogues thus are far richer than in negotiation or auctions. Here agents try to persuade by threatening, by appealing to authority or by promising a reward in the future, just to mention a few possibilities. Figure 2.3 shows a protocol for argumentative dialogues.

Again, in this case, the language gets more complex. Not only the set of illocutionary particles needs to be expanded to include the likes of *threaten*, *reward* or *appeal* [24], but the content of the illocutions has to contain arguments to try and convince our opponent of, for instance, the impossibility of accepting a proposal or the preferability of certain attribute values over others, as well as permitting the critique of certain aspects of a proposal. Broadly speaking agents need to use a language rich enough to build arguments. Propositional logic, or even better first order logic seems necessary for this purpose. The language gets more complex for a more complex problem: *distributed search for an agreement (in multidimensional spaces) with dynamically changing preference sets.*

The decision procedures to be used by agents are again, in this case, more complex. On top of the action choices that an agent already has in negotiation – such as, for example: "how much do I concede", "do I wait?", "should I trade-off issues a and b, or do I better concede?"– there is a complete new set of possible decisions to consider: "should I trade-off or should I argue?", "what argument should I send?", "should I argue in favour of my last proposal?", "should I attack my opponent's last argument or should I send a new argument in favour of my last proposal?". The decisions are more complex because we have the possibility of using the current search space to proceed on the search (as in negotiation) and we have the new possibility of *changing* the search space by arguing, so that offers become interesting to the opponent.

In terms of language complexity, in argumentation we observe a richer set of illocutionary particles, and richer pragmatics as well. Also, the content language gets necessarily enriched by the fact that the some of the new illocutionary acts involve arguments or explanations, that in several cases will have reflexive capabilities to refer to previous arguments or previous offers. A much more complex language scenario for a far more complex problem to be solved.

In all these three cases, even in the simplest one, the underlying context of the negotiation is far more complex than we have been able to describe in terms of the negotiation mechanisms, and the agent interactions involved require some sort of social structure to properly support the interaction conventions, as we will argue next.

BT quoting We shall exemplify argumentative dialogues with an illustrative scenario motivated by work in the ADEPT project [8] which developed negotiating agents for business process management applications. In particular, the researchers considered a multi-agent system for managing a British Telecom (BT) business process —namely, providing a quotation for designing a telecommunications network which offers particular services to a commercial customer. The overall process receives a customer service request as its input and generates as its output a quote specifying how much it would cost to build a network to realise that service. Here we consider a subset of the agents involved in this activity: the customer service division (CSD) agent, the design division (DD) agent, the surveyor department (SD) agent, and the various agents who provide the out-sourced service of vetting customers (VC agents). A full account of all the agents and their negotiations is given in [23].

In order to properly model this problem domain, we found that agents would need to exchange, at least, three types of argument. These are: *threats* (failure to accept this proposal means something negative will happen to you), *rewards* (acceptance of this proposal means something positive will happen to you), and *appeals* (you should prefer this option over that alternative for some reason). Not surprisingly, these argument types are also amongst the most common in the general argumentation literature [9,26].

Figure 3 illustrates the argumentative dialogue between DD and SD agents. The following list enumerates the arc labels.

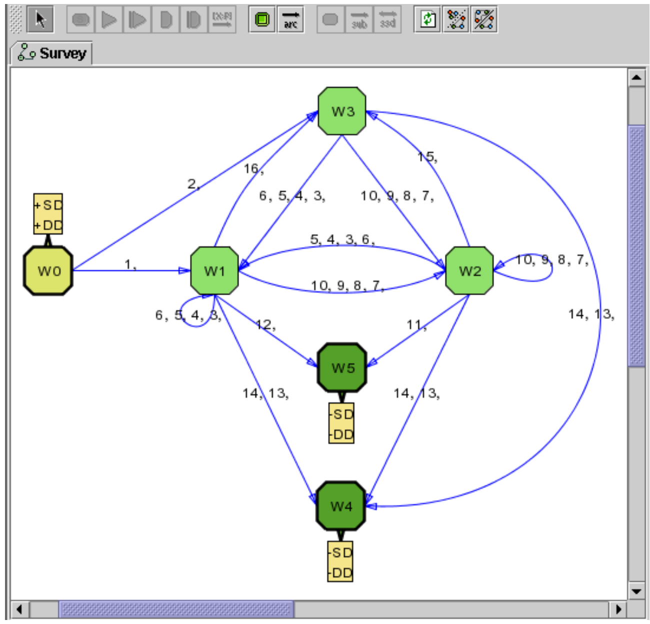

Fig. 3. Argumentation protocol example. The protocol corresponds to an argumentative protocol between agents DD and SD.

1 $\mathit{offer}(?a : DD, ?b : SD, ?\mathit{wff})$

2 $\mathit{request}(?a : DD, ?b : SD, ?\mathit{wff})$

3 $\mathit{offer}(!a : DD, !b : SD, ?\mathit{wff})$

4 $\mathit{threaten}(!a : DD, !b : SD, ?\mathit{wff})$

5 $\mathit{reward}(!a : DD, !b : SD, ?\mathit{wff})$

6 $\mathit{appeal}(!a : DD, !b : SD, ?\mathit{wff})$

7 $\mathit{offer}(!b : SD, !a : DD, ?\mathit{wff})$

8 $\mathit{threaten}(!b : SD, !a : DD, ?\mathit{wff})$

9 $\mathit{reward}(!b : SD, !a : DD, ?\mathit{wff})$

10 $\mathit{appeal}(!b : SD, !a : DD, ?\mathit{wff})$

11 $\mathit{accept}(!a : DD, !b : SD, ?\mathit{wff})$

12 $\mathit{accept}(!b : SD, !a : DD, ?\mathit{wff})$

13 $\mathit{withdraw}(!a : DD, !b : SD)$

14 $\mathit{withdraw}(!b : SD, !a : DD)$

$$15 \quad \mathit{reject}(!a : DD, !b : SD, ?\mathit{wff})$$
$$16 \quad \mathit{reject}(!b : SD, !a : DD, ?\mathit{wff})$$

3 Electronic Institutions.

3.1 Intuitive Notions

Human interactions very often follow *conventions*; that is, general agreements on language, meaning, and behaviour. By following conventions, humans attempt to make their interactions more effective by decreasing uncertainties in the behaviour of others, removing conflicts on meaning, having clearer expectations on the possible outcomes of the interaction and, in general, simplifying the decision process by restricting the potential actions that may be undertaken to a limited set. These benefits explain why conventions –of different sorts– have been so widely used in various domains of human interaction such as trading, public service, games, and the like.

In some situations, conventions become foundational and, in a sense, those conventions become *norms* [2] that establish how interactions of a certain sort will and *must* be structured within an organisation. Such is the essence of what is commonly understood as a human institutions [15]. Human institutions not only structure human interactions, but they are also supposed to enforce individual and social behaviour by obliging every one involved to act according to its norms [20]. This is the case of, for instance, auction houses, courts of law, parliaments or stock exchanges where, in order to achieve a goal, participants –be they staff members of the institution or external participants– must behave within the institution according to the explicit conventions of that institution.

The benefits derived in human organisations by establishing and following conventions become even more apparent when we move into an electronic world where human interactions are mediated or carried out by software agents [12]. In such cases, conventions are necessary to avoid conflicts in meaning, to structure interaction protocols, and to limit the action repertoire of participants, much in the same way that human institutions work; however, the electronic world is a setting where participants may be software entities endowed with limited rationality. The notion of *electronic institution* thus becomes a natural extension of human institutions by permitting not only humans, but also autonomous agents, to interact with one another.

Considering the computer realisation of an institution, we take the stance that *all* interactions among agents are realised by means of message interchanges. We take a strong dialogical stance in the sense that we view multi-agent systems as a type of *dialogical system* [17,13]. The interaction between agents within an electronic institution therefore becomes an exchange of illocutions. In accordance with the classic understanding of illocutions (e.g. Austin [1] or Searle [22]), illocutions "change the world" by establishing or modifying the commitments or obligations of the participants. Therefore, formally speaking, an agent in an electronic institution is any entity capable of establishing commitments. In other words, an entity not capable of establishing commitments should not (must not) speak in an institution. The notion of being capable of establishing commitments is the cornerstone of the construction of institutions because otherwise no notion of enforcement or penalty could arguably be used. In a sense, institutions exist because they are the warrants of the satisfaction of the commitments established by the participants.

In the next subsection we describe the various types of convention that, we think, need to be established in order to properly specify an electronic institution [13,19].

3.2 Electronic Institutions basic concepts

Electronic institutions, as well as human institutions deal with two complementary aspects of the conventions that articulate interactions. On one hand, they need to make explicit for participating agents what is significant (or pertinent) in such interactions and therefore institutions need to address *ontological* (and contextual) issues that are to be used by participating agents in their illocutory exchanges, with a shared meaning and with specific intended effects. On the other hand, institutions need to make explicit those *deontological* aspects that govern the accepted behaviour of participants and may in the end warrant the satisfaction of commitments made within the institution. In order to address these aspects, we have found useful to organize the conventions that govern electronic institutions into three types: ontological and communication conventions, social conventions that govern collective interactions, and rules that normalise (mostly) individual behavior. Each of these three types of conventions are discussed immediately below in a separate section (for a formal account of these components see [13,19,18]).

Ontological and Communicational Conventions: The Dialogical Framework

These conventions help to clarify the meaning of the illocutions exchanged among participants. In order to do so, one needs to make explicit what are the entities the institution deals with, that is, the goods, participants, roles, locations, time intervals, and so on [13]. Likewise, the precise language for interaction should become explicit as part of these conventions. To this end, we propose that communicating agents must share a *dialogical framework* [14]. This is composed of a communication language, a representation language for domain content and an ontology. By sharing a dialogical framework, heterogeneous agents can exchange knowledge and co-ordinate actions with one another.

Definition 1. A *dialogical framework DF* is a tuple $\langle R, SS, O, L, I, CL, Time\rangle$ where

- R stands for a role set (or a set of accepted roles);
- $SS \subseteq R \times SR \times R$ stands for a social structure with SR a set of social relations identifiers;

- O stands for an ontology (vocabulary);
- L stands for a representation language for domain content;
- I is the set of illocutionary particles;
- CL is the (agent) communication language;
- $Time$ is a discrete and partially ordered set of instants.

The role set provides an abstract characterization of the functioning of the different agents in the system [16]. The representation language (e.g., KIF [6] or first-order logic) allows the encoding of the knowledge to be exchanged among agents using the vocabulary offered by the ontology O. The ontology contains constants and terms relative to the domain, including predicate identifiers like *sold*, *withdrawn*, *collision*, or *startingprice*, and constants like *cod*, *20USD*, or *Titanic*. Propositions built with the aid of L, the "inner" language, are embedded into an "outer language", CL, which expresses the intentions of the utterance by means of the illocutionary particles in I. A possible set of illocutions could be {assert, not_assert, request, declare, offer, deny, accept, command }. We take this approach in accordance with speech act theory which postulates that utterances are not simply propositions that are true or false, but attempts on the part of the speaker that succeed or fail [22]. We consider that CL expressions are constructed (like in KQML) as formulae of the type $\iota(\alpha_i : \rho_i, \alpha_j : \rho_j, \varphi, \tau)$ where $\iota \in I, \alpha_i$ and α_j are terms which can be either agent variables or agent identifiers, ρ_i and ρ_j are terms which can be either role variables or role identifiers, $\varphi \in L$ and τ is a term which can be either a time variable or a value in $Time$.

To illustrate these concepts consider the following dialogical framework for a bidding round of a Dutch auction in a fish market (previously described in 2.1). The set of roles is

$$
\begin{aligned}
R = \{&boss, auctioneer, buyer\ \ manager, buyer\ \ admitter, seller\ \ manager,\\
&seller\ \ admitter, buyer, seller\}\\
SS = \{&(boss, has\_power, auctioneer),\\
&(boss, has\_power, buyer\ \ manager),\\
&(boss, has\_power, buyer\ \ admitter),\\
&(boss, has\_power, seller\ \ manager),\\
&(boss, has\_power, seller\ \ admitter),\\
&(auctioneer, has\_authority, buyer),\\
&(auctioneer, has\_authority, seller)\}
\end{aligned}
$$

The chief staff agent of the auction house plays the role of *boss*. It exerts power over all agents playing the other staff roles of the institution: *auctioneer* responsible of the actual fish auctioning and the different buyer and seller *managers* and *admitters* responsible of payments to and from sellers and buyers and of the admittance of participants respectively. Auctioneers have authority over sellers and buyers because the boss delegates power to them about decisions on winner determination and on modification of auction conditions (order of auctioning, withdrawal of products, etc.).

Social Conventions: Scenes and Performative Structure

These conventions regulate the interactions among participants. They contain agreements on protocols and on the sequence of activities in the institution. *Scenes* regulate the protocol to follow for each individual activity and the *Performative Structure* establishes the links and traversal paths between the scenes.

The overall activity within an electronic institution is a composition of multiple, well-separated, and possibly concurrent, dialogical activities, each one involving different groups of agents playing different roles. For each such activity, which we will call a scene, interactions between agents are articulated through the meeting of various groups of agents that follow well-defined communication protocols. Thus, for example, in the context of an auction house, there are the following scenes involving the following agents and the following protocols: buyer and seller admission scenes subject to an information seeking protocol (staff members acquire information concerning the goods to be auctioned and the credit of buyers), an auction scene subject to the corresponding auction protocol (for instance, English or Dutch), and scenes corresponding to the buyer and seller settlements that correspond to contract signing protocols (buyers get the goods in return for money and sellers get money in return for the goods that have been sold). In fact, with this model no agent interaction can take place outside of the context of a scene. We consider the protocol of each scene to model the possible dialogical interactions between the group of agents playing specific roles. In other words, scene protocols are patterns of multi-role conversation [13].

A scene protocol is specified by a graph whose nodes represent the different states of the conversation and the arcs connecting the nodes are labelled with illocutions that make the scene state change (see Figure 3). The graph has a single initial state (non-reachable once left) and a set of final states representing the different acceptable endings of the conversation. There is no arc connecting a final state to some other state.

Figure 3 shows an example of a scene protocol specified with the ISLANDER toolbox [3]. This scene corresponds to the `Survey` scene of the institution presented in Figure 6. Normally, the correct evolution of a conversation protocol requires a certain number of agents for each of the various roles involved in it, in the example in Figure 3 we have two roles a Surveyor Department agent (SD) and a Design Division agent (DD). The set of roles will be denoted by the symbol R. Then a minimum and maximum number of agents per role is defined and the number of agents playing each role has to be in this interval –in this example, it happens to be one agent both as minimum and maximum– (denoted by two functions *min* and *Max*). Because we need to model multi-agent conversations in which the set of participants may dynamically vary, scenes need to be specified such that agents can either join in or leave at particular moments during an ongoing conversation. For this purpose, we differentiate the sets of *entrance* (denoted by *WA*) and the *exit* states (denoted by *WE*) for different roles. The entrance or exit of agents has to satisfy the restriction mentioned above about the number of agents for each role. Obviously, the final states ought to have exit states for each role, in order to allow all the agents to leave when the scene is finished. In contrast, the initial state has to be an access state for the roles whose minimum is greater than zero, in order to start the scene.

Definition 2. Formally, a scene is a tuple[3] where:

$$S = \langle R, W, w_0, W_f, (WA_r)_{r\in R}, (WE_r)_{r\in R}, \Theta, \lambda, min, Max\rangle$$

- R is the set of roles of the scene, a subset of R in Definition 1;
- W is a finite, non-empty set of scene states;
- $w_0 \in W$ is the initial state;
- $W_f \subseteq W$ is the non-empty set of final states;
- $(WA_r)_{r\in R} \subseteq W$ is a family of non-empty sets such that WA_r stands for the set of access states for the role $r \in R$;
- $(WE_r)_{r\in R} \subseteq W$ is a family of non-empty sets such that WE_r stands for the set of exit states for the role $r \in R$;
- $\Theta \subseteq W \times W$ is a set of directed edges;
- $\lambda : \Theta \longrightarrow CL$ is a labelling function relating each transition with an illocution schema expressed in the language CL.
- $min, max : R \longrightarrow I\!N$ $min(r)$ and $max(r)$ return respectively the minimum and maximum number of agents that must and can play the role $r \in R$;

Notice that not every illocution scheme is valid to label an arc. In general, a CL expression $\iota(\alpha_i : \rho_i, \alpha_j : \rho_j, \varphi, \tau)$ from Definition 1 can label an arc if:

- α_i and α_j are agent variables;
- ρ_i and ρ_j are either role variables or role identifiers in R_s; and
- τ is a time variable;

These variables will be bound to concrete values during the execution of the scene. For example, agent variables in an illocution scheme will be bound, respectively, to the identifier of the agent that has uttered the illocution and to the identifier of the agent who has received the illocution. Then at each moment, the bindings of the variables will be the context of the scene execution. These variables have a local scope within a scene execution that, as said, represents an actual activity undertaken by a group of agents. There are several such activities within any possible institution, so there must be a way of modelling their interconnections. The notion of a *performative structure* is the most complex and interesting of the proposed formalism, since it precisely models the relationships among scenes.

Notice that although conversations (scenes) are currently admitted as the unit of communication between agents, limited work has been done concerning the modelling of the relationships between different scenes. This issue is particularly significant when conversations are embedded in a broader context, such as, for instance, organisations and a hierarchy of institutions. If this is the case, it is

[3] When we need to differentiate the elements of two scenes s and s' we will use a superindex s or s'.

important to capture the relationships between scenes. Our argument is that this is precisely the main difference with classic Mechanism Design, understood with the narrow view of the description of the central activity of an interaction. We believe that for MAS specification, a specification model that has a holistic view of the pragmatics and the web of activities (central and peripheral) is essential. Electronic Institutions aim at playing this role.

In general, the activity represented by a performative structure can be depicted as a collection of multiple, concurrent scenes. Agents navigate from scene to scene constrained by the rules defining the relationships between scenes. Moreover, the same agent can potentially participate in multiple scenes at the same time. From a structural point of view, performative structures' specifications must be regarded as networks of scenes. At execution time, a performative structure becomes populated by agents that make it evolve whenever these comply with the rules encoded by the specification. Concretely, an agent participating in the execution of a performative structure devotes its time to jointly start new scene executions, to enter active scenes where the agent interacts with other agents, to leave active scenes to possibly enter other scenes, and finally to abandon the performative structure.

At this point it should be noted that the way agents move from scene to scene depends on the type of relationship holding between the source and target scenes. Sometimes we might be interested in forcing agents to synchronise before jumping into either new or existing scene executions, or offering choice points so that an agent can decide which target scene to incorporate itself into, and so on. Summarizing, in order to capture the type of relationships listed above, we consider that any performative structure can contain special elements (that we call *transitions*) whose function is to mediate different types of connections among scenes. Each scene may be connected to multiple transitions, and in turn each transition may be connected to multiple scenes. In both cases, the connection between a scene and a transition is made by means of a directed arc. Then we can refer to the source and target of each arc. And given either a scene or a transition, we shall distinguish between its incoming and outgoing arcs. Notice that there is no direct connection between two scenes (i.e., all connections between scenes are mediated by transitions). Also we do not allow the connection of transitions. Each arc connecting a scene with a transition is labelled with the roles played by the agents that traverse it and a set of constraints that must be satisfied by the agents. Any agent playing a different role from those marked on the arc or not satisfying the constraints will not be authorised to abandon the scene at the beginning of the arc. Similarly, arcs connecting a transition with a scene are labelled with the roles that the agents traversing it will play in the target scene. See an example of performative structure in Figure 6.

Agents move from a scene instance (execution) to another by traversing the transition connecting the scenes and following the arcs that connect transitions and scenes. Transitions should therefore be regarded as a kind of router that contains local information about the scene instances that they connect. Therefore, instead of modelling some activity, they are intended to route agents towards their destinations in different ways, depending on the type of transition. The arcs connecting transitions to scenes also play a fundamental role. Notice that as there might be multiple (or perhaps no) scene executions of a target scene, it should be specified whether the agents following the arcs are allowed to start a new scene execution, whether they can choose a single or a subset of scenes to incorporate into, or whether they must enter all the available scene executions. Formally:

Definition 3. A performative structure is a tuple

$$PS = \langle S, T, s_0, s_\Omega, E, f_L, f_T, f_E, C, \mu, ML \rangle$$

where

- S is a finite, non-empty set of scenes; defined according to Definition 2.
- T is a finite and non-empty set of transitions;
- $s_0 \in S$ is the *root* scene;
- $s_\Omega \in S$ is the *output* scene;
- $E = E^I \cup E^O$ is a set of arc identifiers where $E^I \subseteq S \times T$ is a set of edges from scenes to transitions, and $E^O \subseteq T \times S$ is a set of edges from transitions to scenes;
- $f_L : E \to 2^{V_A \times R}$ is the labelling function associating each arc with pairs of agent variables and roles;
- $f_T : T \to \tau$ maps each transition to its type — where $\tau = \{sync/parallel, choice/choice, sync/choice, choice/parallel\}$ corresponds to the behaviour of the transition with respect to the incoming and outgoing arcs;
- $f_E : E^O \to \epsilon$ maps each arc from transition to scene to its type - where $\epsilon = \{1, some, all, new\}$ correspond to one, several, all, or a newly created running execution of the target scene respectively;
- $C : E \to ML$ maps each arc to a meta-language expression representing the arc's constraints that agents must satisfy to follow the arc;
- $\mu : S \to \mathbb{N}$ sets an upper bound to the number of allowed simultaneous running executions of a given scene; and
- ML is a meta-language over CL and L as defined in Definition 1.

Behaviour Conventions: Normative Rules

These conventions determine the socially pertinent commitments for the participating agents and describe their various obligations and rights. As discussed so far, a performative structure can be seen to constrain an agent's behaviour at two levels:

1. *intra-scene:* Scene protocols dictate, for each agent role within a scene, what can be said, by whom, to whom, and when.
2. *inter-scene:* The connections among the scenes, given by the performative structure, define the possible paths that agents may follow depending on their roles. Furthermore, the constraints over output arcs impose additional limitations on the agents when attempting to reach a target scene.

Although these may appear distinct, an agent's actions within a scene may have non-local consequences in that it may either limit or enlarge its acting possibilities in subsequent scenes. Such consequences may have effect along two different directions. On the one hand, some actions will introduce subsequent acting commitments that have to be interpreted as acting obligations. While on the other hand, consequences occurring locally within a scene may vary the paths that an agent can follow in the performative structure because they affect the satisfaction and contravention of the constraints labelling the paths. Both types of consequences need to be kept by an institution on a per agent basis so that the different obligations and restrictions may be subsequently enforced.

In order to represent the deontic notion of obligation (see [27] for background details) we set out the predicate Obl as follows:

$$Obl(x, \psi, s) = \text{agent } x \text{ is obliged to do (in fact, to `say')} \psi \text{ in scene } s. \qquad (1)$$

where ψ is taken to be an illocution scheme. We will note by Obl the set of obligations and by $obl_i \in \mathsf{Obl}$ any concrete obligation.

Behaviour conventions of an Electronic Institution are specified as a special type of rule, called a *normative rule*, that captures which agent actions (illocutions) have consequences that need to be kept in its context. Given a performative structure, the normative rules are written in its meta-language according to the following schema:

$$(s_1, \gamma_1) \wedge \ldots \wedge (s_m \gamma_m) \wedge \neg(s_{m+1}, \gamma_{m+1}) \wedge \ldots \wedge \neg(s_{m+n} \gamma_{m+n}) \rightarrow obl_1 \wedge \ldots \wedge obl_p$$

where $(s_1, \gamma_1), \ldots, (s_{m+n}, \gamma_{m+n})$ are pairs of scenes and illocution schemes, $\neg$ is a defeasible negation, and $obl_1 \wedge \ldots \wedge obl_p$ are obligations. The meaning of these rules is that if the illocutions $(s_1, \gamma_1) \wedge \ldots \wedge (s_m, \gamma_m)$ have been uttered, and the illocutions $(s_{m+1}, \gamma_{m+1}) \wedge \ldots \wedge (s_{m+n}, \gamma_{m+n})$ have *not* been uttered, the obligations $obl_1 \wedge \ldots \wedge obl_p$ hold. Therefore, the rules have two components, the first one is causing the obligations to be activated (for instance winning an auction round by saying 'mine' in a downwards bidding protocol, generates the obligation to pay) and the second is the part that removes the obligations (for instance, paying the amount of money at which the round was won).

We can now show how these ideas are put to work to extend the three examples of agent interaction mechanisms we discussed earlier.

4 Revisiting the mechanisms. The Big Picture

4.1 Auctions

Although Auction houses can be seen as a very simple co-ordination mechanism (in a sense as simple as an ant algorithm), some economists (such as Smith [25] or McAfee and McMillan[11]), however, refer to them as *institutions* [15] and are careful to point out that, in addition to the bidding conventions, other equally relevant conventions and elements are used in an auction house to achieve a proper co-ordination of buyers and sellers: conventions for the registration of participants and goods, conventions on guarantees and payment, commissions, starting prices, etc.

The actual auctioning of goods (what we would call the auctioning scene) can certainly be understood as governed by a simple language and protocol, according to the point of view introduced in Section 2. It is when looking at the global picture of all the activities that take place around that kernel dialogue that we can appreciate that even auctions involve many more and more complex interactions, and are hence supported by complex institutions. Any model that concentrates just on the auction mechanism and ignores the full picture will lead to systems difficult to use in the real world and difficult to integrate with the existing legacy systems. An auction does not end when the auctioneer has the last standing bid in an English auction and assigns the good after the sequence 'going-going-gone' is finished. A process of credit checking, payment procedures, document generation starts that is essential for the successful completion of the transaction.

When analyzing the example of the fish market as introduced in this paper from this broader perspective, we observe a series of activities that complement and modify the simple auction protocol: the buyers' and sellers' registration and settlements. Also, many pragmatic elements have to be modelled (even for this simple mechanism). For instance, the fact that buyers have an associated credit in the auction house changes the winner determination protocol by introducing the fact that the bid of a buyer has to be supported by its current credit, and therefore the auctioneer has to check that. As argued in this paper, even simple mechanisms have complex societal restrictions that require elaborated specification languages. See in Figure 4 the performative structure of the FishMarket.

4.2 Negotiation

In the supply chain example, we have to realise that each chain level is a global view of a reality consisting of many individual agents, and that the transactions modelled as those flows are the result of the social interaction of the agents following particularly well established conventions. For instance, the interaction between the agents at levels S_1 and S_2 is modelled as negotiation. The interaction between levels S_2 and S_3 could probably be better modelled by fixed-price mechanisms. Electronic Institutions offer the necessary principled way to model such interactions. In our Institution specification for the example we consider two scenes: One, *primary_market*, for the interaction between agents of S_1 and S_2 and another, *retailing*, for the interaction between S_2 and S_3. Scene *primary_market* is endowed with a protocol as the one in Figure 2.

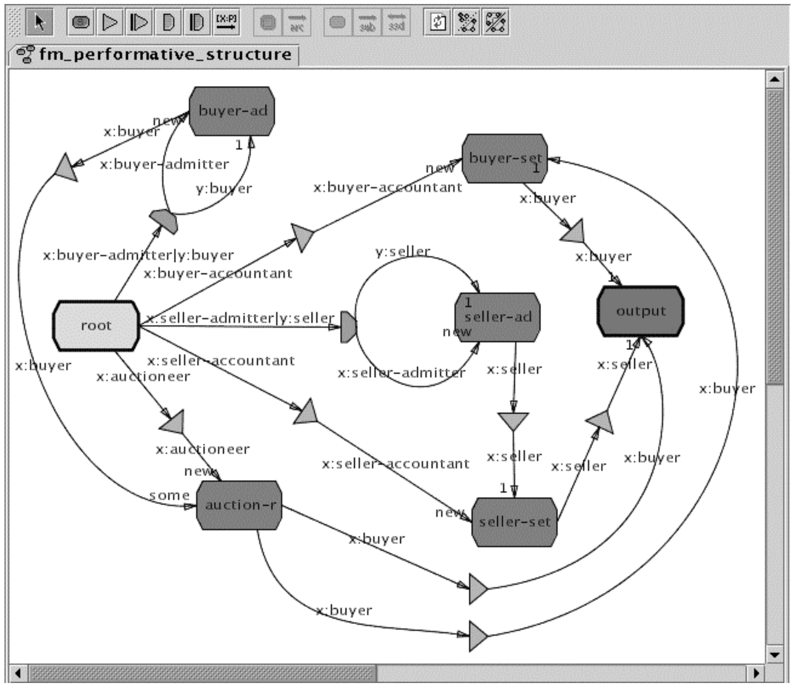

Fig. 4. Performative structure of the FishMarket Institution. Each rectangle represents a scene and the labelled links correspond to the movements of agents between scenes. Transitions are represented as triangles. *root* is the initial scene to enter the institution, and *output* is the scene through which all agents leave the institution.

4.3 Argumentation

In the case of the argumentation example, its electronic institution has to account for many complex activities a part from the scenes where the argumentation takes place. The first stages of the Provide_Customer_Quote service involve the CSD agent capturing basic information about the customer and vetting the customer in terms of their credit worthiness. The latter service is performed by one of the VC agents and negotiation is used to determine which one is selected. If the customer fails the vetting procedure, then the quote process terminates. Assuming the customer is satisfactory, the CSD agent maps their requirements against a service portfolio. If the requirements can be met by a standard off-the-shelf portfolio item then an immediate quote can be offered based on previous examples. In the case of bespoke services the process is more complex. The CSD agent negotiates with the DD agent for the service of costing and designing the desired network service. To prepare a network design it is usually necessary to have a detailed plan of the existing equipment at the customer's premises.

Sometimes such plans might not exist and sometimes they may be out of date. In either case, the DD agent determines whether the customer site(s) should be surveyed. If such a survey is warranted, the DD agent negotiates with the SD agent for the Survey_Customer_Site service. This negotiation differs from the others present in this scenario in that the two agents are part of the same department. Moreover, the DD agent has a degree of authority over SD. Agent negotiation is still required to set the timings of the service, but the SD agent cannot simply refuse to perform the service. On completion of the network design and costing, the DD agent informs the CSD agent which informs the customer of the service quote. The business process then terminates. Figure 6 summarises this institution.

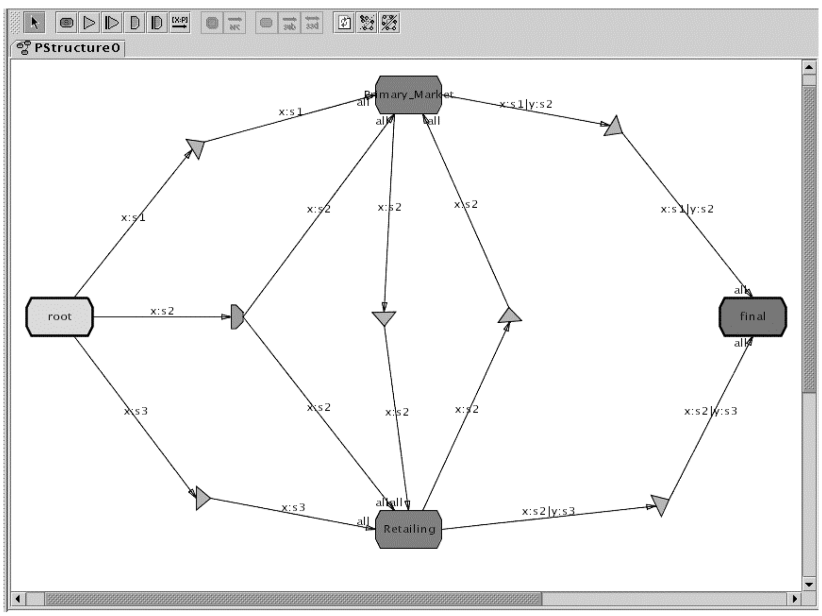

Fig. 5. Performative structure of the Supply Chain Institution.

5 Conclusions

In this paper we argued for the concept of Electronic Institutions as a methodological tool to specify multi-agent systems, as they permit a view that is broader than that offered by the usual mechanism design approach. Electronic Institutions give a handle to designers of multi-agent systems to address the difficult problem of specifying the many inter-related activities required for successful complex agent interactions.

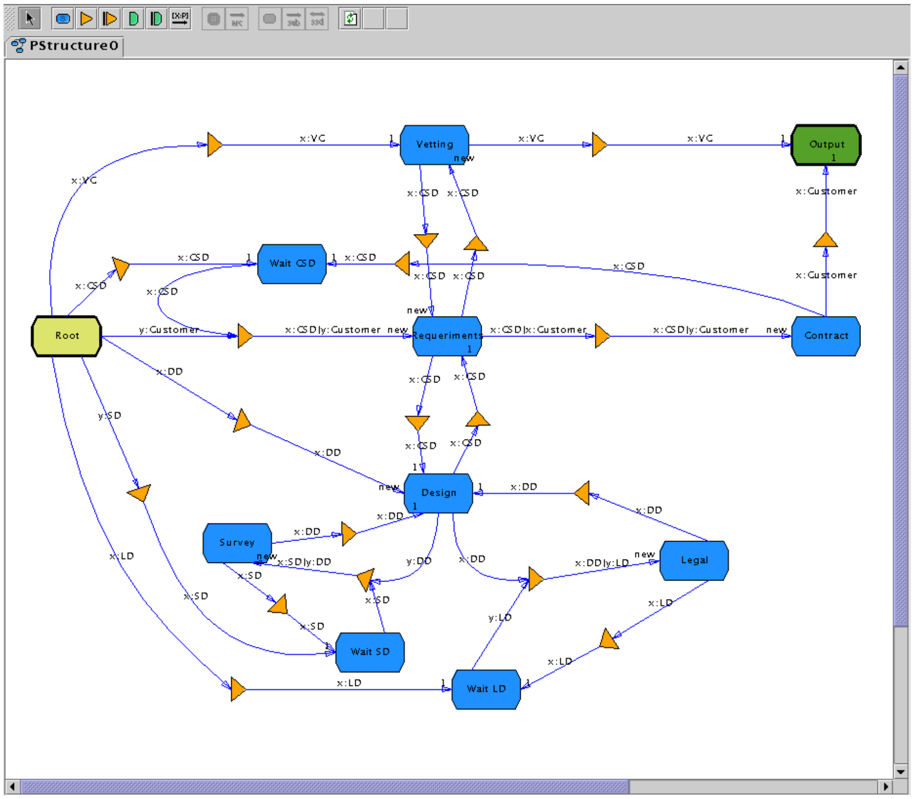

Fig. 6. Performative structure of the BT Institution. As before, each rectangle represents a scene and the labelled links correspond to the movements of agents between scenes. Transitions are represented as triangles.

We also explored the correspondence between the complexity of the problem to be solved and the complexity of the communication language of participating agents. We used three interaction examples –of growing complexity– to illustrate this point, as summarized in the following table:

Interaction	**Pragmatics**	**Content**
Auctions	Offer, Accept	simple terms, integers
Negotiation	Offer, accept	complex terms
Argumentation	Offer, Critique, Appeal, Threaten	complex terms, FOL formulae

6 Acknowledgements

Most of the material of this paper is the result of collaborative work with different researchers among which we'd like to mention Nick Jennings, Simon Parsons, Julian Padget, Juan Antonio Rodriguez, and Peyman Faratin. This line of research is currently being supported by the MCYT research project eINSTITUTOR (TIC2000-1414) and the IST project SLIE (IST-1999-10948). The graphics in the paper have been produced with the help of Marc Esteva using the Electronic Institution specification tool ISLANDER [4].

References

1. J. L. Austin. *How to Do Things With Words.* Oxford University Press, 1962.
2. C. Castelfranchi, F. Dignum, C. Jonker, and J. Treur. *Deliberate Normative Agents: Principles and Architectures*, pages 364–378. Intelligent Agents VI (LNAI-1757). N. Jennings and Y. Lesperance (eds.). Springer-Verlag, 2000.
3. M. Esteva and C. Sierra. ISLANDER1.0 Language Definition. Technical report, IIIA-CSIC, 2001.
4. Marc Esteva, David de la Cruz, and Carles Sierra. ISLANDER: an electronic institutions editor. In *AAMAS'02*, page in press. ACM Press, 2002.
5. P. Faratin, C. Sierra, and N. R. Jennings. Using similarity criteria to make negotiation trade-offs. In *Proc. Fourth Int. Conf. on Multi-Agent Systems, ICMAS'00*, pages 119–126, 2000.
6. Michael R. Genesereth and Richard E. Fikes. Knowldege interchange format version 3.0 reference manual. Technical Report Report Logic–92–1, Logic Group, Computer Science Department, Standford University, June 1992.
7. N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, M. Wooldridge, and C. Sierra. Automated negotiation: Prospects, methods and challenges. *Group Decision and Negotiation*, 10:199–215, 2001.
8. N. R. Jennings, P. Faratin, T. J. Norman, P. O'Brien, and B. Odgers. Autonomous agents for business process management. *Int. Journal of Applied Artificial Intelligence.*, 14(2):145–189, 2000.
9. M. Karlins and H. I. Abelson. *Persuasion.* Crosby Lockwood & Son, London, UK, 1970.
10. B. Laasri, H. Laasri, S. Lander, and V. Lesser. A generic model for negotiating agents. *International journal of Intelligent and Cooperative Information Systems*, 1(2):291–317, 1992.
11. R. P. McAfee and J. McMillan. Auctions and bidding. *Journal of Economic Literature*, XXV:699–738, 1987.
12. P. Noriega and C. Sierra. Auctions and multi-agent systems. In Matthias Klusch, editor, *Intelligent Information Agents*, pages 153–175. Springer, 1999.
13. Pablo Noriega. *Agent-Mediated Auctions: The Fishmarket Metaphor.* Number 8 in IIIA Monograph Series. Institut d'Investigació en Intel.ligència Artificial (IIIA), 1997. PhD Thesis.
14. Pablo Noriega and Carles Sierra. Towards layered dialogical agents. In *Third International Workshop on Agent Theories, Architectures, and Languages, ATAL-96*, 1996.
15. Douglas C. North. *Institutions, Institutional Change and Economics Perfomance.* Cambridge U. P., 1990.

16. P. Panzarasa, T. J. Norman, and N. R. Jennings. Social mental shaping: modelling the impact of sociality on autonomous agents' mental states. *Computational Intelligence*, 2001.
17. N. Rescher. *Dialectics: A controversy-oriented approach to the theory of knowledge.* SUNY, 1977.
18. Juan A. Rodríguez-Aguilar, Francisco J. Martín, Pablo Noriega, Pere Garcia, and Carles Sierra. Towards a test-bed for trading agents in electronic auction markets. *AI Communications*, 11(1):5–19, 1998.
19. Juan Antonio Rodriguez-Aguilar. *On the design and construction of Agent-mediated Institutions.* Number 14 in Monografies de l'IIIA. IIIA-CSIC, 2002.
20. L. Royakkers and F. Dignum. Organisations and collective obligations. In *11th International Conference on Databases & Expert Systems Applications (LNCS-1873)*, pages 302–311. Springer-Verlag, 2000.
21. N. Sadeh-Koniecpol, D. Hildum, D. Kjensta, and A. Tseng. Mascot: An agent-based architecture for coordinated mixed-initiative supply chain planning and scheduling. In *Workshop notes, Agent-Based Decision Support for Managing the Internet-Enabled Supply Chain, Third International Conference on Autonomous Agents (Agents '99)*, 1999.
22. J. R. Searle. *Speech acts.* Cambridge U.P., 1969.
23. C. Sierra, P. Faratin, and N. R. Jennings. A service-oriented negotiation model between autonomous agents. In *MAAMAW'97*, number 1237 in LNAI, pages 17–35, Ronneby, Sweden, 1997.
24. C. Sierra, N. R. Jennings, P. Noriega, and S. Parsons. A framework for argumentation–based negotiation. In *Intelligent Agents IV*, pages 177–192, 1997.
25. Vernon L. Smith. *Auctions*, pages 39–53. The new Palgrave: a dictionary of Economics. John Eatwell, Murray Milgate and Peter Newman (eds). McMillan, London, 1987.
26. K. P. Sycara. Persuasive argumentation in negotiation. *Theory and Decision*, 28:203–242, 1990.
27. G. H. von Wright. Deontic logic. *Mind*, (60):1–15, 1951.