Interaction Patterns and Observable Commitments in a Multi-Agent Trading Scenario

Jeremy Pitt, Lloyd Kamara and Alexander Artikis

Intelligent & Interactive Systems Group
Department of Electrical & Electronic Engineering
Imperial College of Science, Technology & Medicine
Exhibition Road, London, SW7 2BT, UK
Email: {j.pitt,l.kamara,a.artikis}@ic.ac.uk URL: http://www.iis.ee.ic.ac.uk/

Abstract

We propose a formal semantics for the protocol diagrams (interaction patterns) of AUML (Agent Unified Modelling Language). We connect this proposal with a general framework for defining the semantics of ACLs (Agent Communication Languages). We then show that protocol diagrams should be parameterised with observable commitments: additional specification of the expected outcomes, norms and obligations resulting from the use of the protocol. A complete axiomatisation of a contract-net protocol is given, and animated to show how the agents comply with expected replies and respecting the norms. We conclude that this approach to 'socialising' interaction between agents is important for developing open agent systems and potentially useful in standardisation.

1 Introduction

In previous work, we defined the semantics of speech acts in an Agent Communication Language (ACL) from two perspectives. From the external point of view, we were concerned with performatives and protocols, and defined the semantics of a speech act as an intention to make a reply [12]. From the internal point of view, we were concerned with the interpretation of content and the mental attitudes motivating the (intention to) reply, which were given an operational semantics through a transformation into Prolog [10]. Our primary concern was the intentional specifications of behaviour using a BDI (Beliefs-Desires-Intentions) agent model. We therefore used finite state diagrams in our representation of protocols, for economy of representation, ease of understanding, and a semantics that was compatible with the general semantic framework for ACLs.

The agent standards organization FIPA (Foundation for Intelligent Physical Agents) has focused in its specifications on the protocols [6], and is using AUML (Agent Unified Modelling Language [1]) for this purpose. Unfortunately, the form of graphical protocol representation proposed does not have a formal semantics, only English descriptions. In design, the potential ambiguity can lead to inconsistency of interpretation and, at implementation, failure to interoperate. There is also no specification of the intended decision-making processes and no representation of the *point* of the protocol, i.e. what it is intended to achieve.

In this paper, we move towards a formal semantics for AUML protocol diagrams and, in so doing, embed these representations in our general semantic framework. In addition, we use ideas from [9] to parameterise the specifications with the norms that govern the use of the speech acts in the context of the protocol, and the observable commitments (outcomes and obligations) that are produced as a result of following the protocol to a successful conclusion. The processes and parameters specified in the protocol are captured in the intentional specifications of agent behaviour.

The argument followed in this paper is then as follows:

- we start from the description of a multi-agent trading scenario and the design of an ACL for the interactions;
- the interactions are specified using AUML protocol diagrams, which are given a semantics that is embedded in the general semantic framework for ACLs,
- the graphical representation is then associated with intentional specifications of agent behaviour;
- the protocol diagram (interaction pattern) is parameterised with observable commitments, i.e. norms that the agents should comply with;
- the specification is animated to show that the agents comply with all 'externals', including making the appropriate kind of reply and complying with the norms.

We conclude that this use of AUML is a potentially powerful enhancement to our general semantic framework and ACL design methodology [10] (although we discuss some reservations in Section 7.3. It is therefore a significant contribution to standardization efforts, but also advances our objectives in the EU ALFEBIITE project. This is to investigate the use of normative and norm-governed behaviour in communications between agents, in order to create open agent societies.

2 Motivation

We share a vision of a flexible network of heterogeneous software processes (i.e. independently designed and implemented agents) coming together to form an open agent society. By this we mean:

- *open*: high-level interoperability ('public' accessibility) and an unpredictable, non-deterministic environment;
- *agent*: local autonomy, adjustable behaviour and high-level communication
- *society*: the collection of agents is regulated by the kinds of relations contractual and normative) found in human business and social situations.

In particular we are interested in how such agent societies facilitate future commercial and social structures, like Connected Communities and Virtual Enterprises.

To investigate these ideas, inter alia, the EU ALFEBIITE project has developed a number of scenarios to investigate norm-governed trading in multi-agent systems. We have specified one scenario based on an abstract producer–consumer model, in which there are a number of agents producing 'goods', a number of agents consuming goods, and transactions between the two sorts of agent which can be broken by either side. In previous work, we analysed quantitative representations of trust in an instance of this scenario using Intelligent Networks [16].

A new instantiation of this scenario is based on cartography and exploration. In this case, we specify a set of explorer agents, who can generate raw data, and a set of cartographer agents, who want to build a complete map. (Our working example is oil exploration, with explorer agents getting seismographic reading and cartographer agents compiling sets of such readings to build a complete picture to identify oil deposits). The important features of this scenario (illustrated in figure 1) are:

- *mixed initiative*: cartographers can use the contract-net protocol to contract a particular explorer to search a certain region, while explorer agents can put pro-actively explored regions up for auction. Interaction can be broadcast or multicast;

- *third parties*: there are many third parties involved in the above interactions, including an auction house, bank, certification authority, rights managers, etc.;

- *rights*: there are issues of rights and ownership, e.g. granting of a right to explore a region, and ownership of information (i.e. data that has been contracted for can't be offered at auction). Rights are specified in a contract containing a service level agreement;

- *norms*: the interaction between agents follow norms and can create new ones, in the form of permissions, obligations, powers, etc. There are also social relations involved, for example, trust, control, and reputation;

- *visualization*: the interactions, contracts and social relations between agents can be visualised to make what is conceptually relevant perceptually prominent.

In this paper, we are concerned with designing an Agent Communication Language for this scenario. However, in related and further work, we are (will be) concerned with formal definition of the artificial society, experiments with trade-offs between trust and control (cf. [2]), and with investigation of issues such as scale, stability and self-regulation Note that as outlined here, some of the agent internals (for example, the bidding and contracting strategies to maximise financial advantage) are not of direct concern.

3 Agent UML

In this section, we briefly review the protocol diagram notation for agent interaction protocols proposed in [1] and used in the FIPA specifications [6]. We then give an illustrative example using the well-known contract-net protocol.

3.1 Overview of AUML

Agent Unified Modelling Language (AUML) is a proposed extension of UML for multi-agent systems, on the grounds that agents require additional, richer, modelling techniques than objects [1]. AUML has been adopted as part of the FIPA specifications for defining a standard Agent Communication Language FIPA-ACL [6].

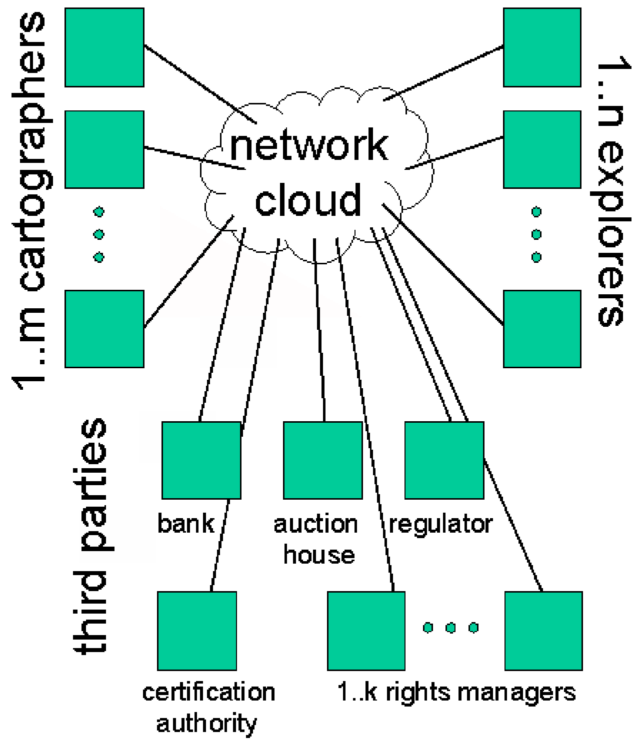

Figure 1: Multi-Agent Trading Scenario

At the core of AUML is a mechanism for describing the interactions between agents using protocol diagrams. Protocol diagrams are therefore concerned with defining the allowed sequence(s) of messages exchanged between agents for some common purpose. A graphical notation is used, as illustrated in figure 2. The notation is very general and powerful, allowing the designer to specify a wide range of complex interactions. The specification devices supported include:

- *lifelines*: the time period during which the agent is active in its *role* in the protocol;
- *roles*: agents satisfying certain properties, performing particular actions, or capable of specified behaviours in a protocol;
- *threads of interaction*: showing the period during which an agent is performing some task in reaction to a received message;
- *communicative acts*: the type of message exchanged between agent (roles), parameters, and other options (synchronous, asynchronous, etc.);
- *parallelism*: and-, or-, and x-or parallelism between communicative acts;
- *guards*: conditions on performing communicative acts;
- *cardinality*: support for one-one, one-many, etc. interactions;
- *templates*: allowing protocols to be parameterised to define a class of protocols.

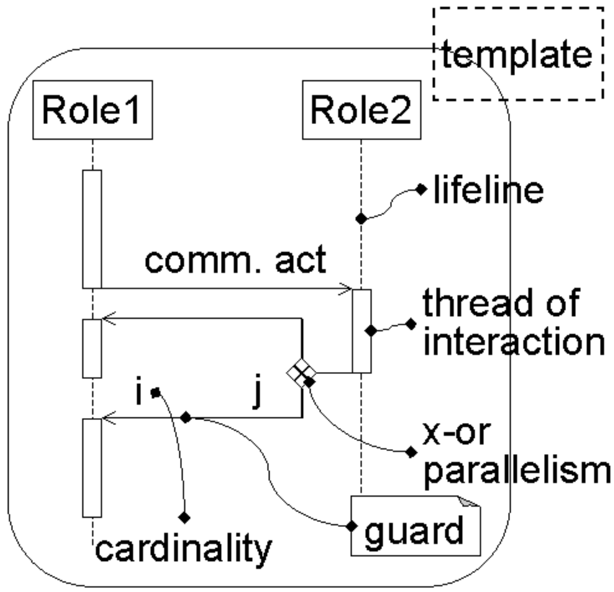

Figure 2: Example AUML Protocol Diagram

3.2 Contract-Net Protocol

The AUML protocol diagram notation is very descriptive, although this comes at the cost of perspicuity. For example, a contract net protocol and an auction protocol are required in the multi-agent trading scenario proposed in Section 2. An English auction protocol is specified in [1]. Although this starts with a *cfp* performative ('call for proposals'), presumably this could just be re-used in our scenario. However, the specification in [1] (p210) is somewhat confusing and seemingly at variance with the English description that follows it[1] A rather more convincing specification of an English auction protocol can be found in [11].

A contract-net protocol can be found in the FIPA specifications [6], although this too turns out to be not ideal for our requirements. Intuitively, we specify our protocol as follows:

the contractor (agent role) sends a cfp to n bidders;
contractor receives m ($m \leq n$) responses, of which:
  i are reject, which ends the protocol,
  k are accept, received after a timeout, to be rejected,
  j are accept, received before the timeout;
the contractor selects a winner from the set j;
the contractor then sends:
  $j - 1$ rejects to the unsuccessful bidders,
  an accept to the contract winner;
the winner then performs the contracted task;

[1] Confusions include: mixing auction set-up and execution phases; mixing message types and content; reversing time flows; ambiguity over message cardinality; ambiguity over message cardinality and x-or parallelism; following a not-understood by an inform(end of auction); "as soon as 1" in the English description not captured in the diagram; no representation of timeout to end auction; no representation of 'silence' or no response from an agent.

after performing the task it notifies the contractor.

The protocol diagram for this contract-net protocol is shown in Figure 3. Note that errors in understanding, syntax, and the communications fabric have not been considered. Also, the content of the messages and the decision making are also not specified. We are concerned here only with the *form* and pattern of exchanged message from the external point of view. Agent *internals* – intentions and interpretations – will be dealt with at the next level of specification, as discussed in section 5.

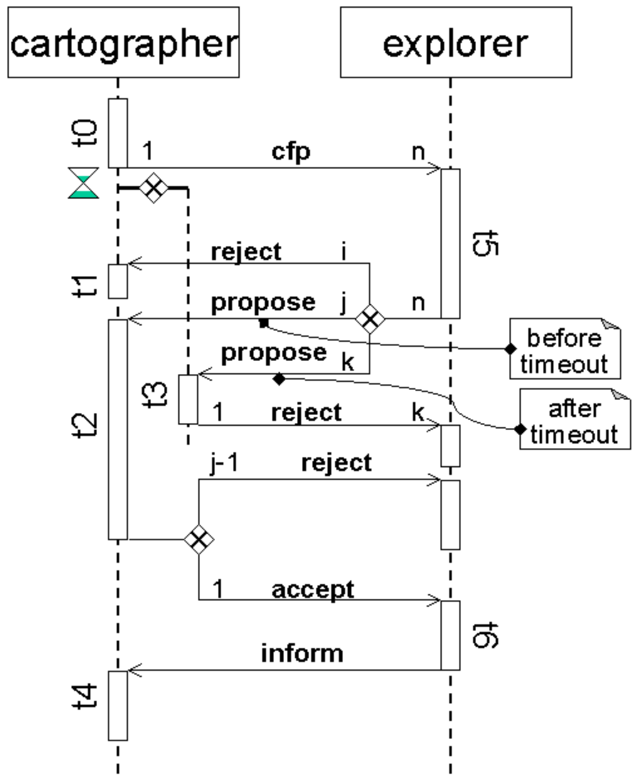

Figure 3: Contract-Net Protocol in AUML

4 Protocol Diagram Semantics

The AUML protocol diagram themselves do not have a formal semantics. Furthermore, the behaviour of the agents in following the protocol is not specified: 'the protocol says nothing about how the reaction [reply to a received message] is implemented" [1]:p211. However, implementation is not the issue: a specification of the decision-making is required for designers to implement the protocol according to its intended effect. The intended achievement of the protocol (i.e. what the agents are communicating *about* or *for*) is also not specified in the protocol diagram.

In this section we address the the first of these three deficiencies. We briefly review our general semantic framework for completeness (for full details, see [12, 10]. We then propose a semantics for the AUML protocol diagram which can be linked with this framework. In section 5 we show how the decision-making required in threads of interaction can be specified using a BDI-style logical language. Finally we show how the protocol should be parameterised by norms which condition the use of speech acts in the context of the protocol and the obligations that follow as a result.

4.1 General Semantic Framework for ACLs

We define an ACL to be a 3-tuple $< Perf, Prot, reply >$ where *Perf* is a set of performative names, *Prot* is a set of protocol names, and *reply* is a partial function given by:

$$reply : Perf \times Prot \times \mathrm{N}^+ \longmapsto P(Perf \times Prot)$$

where N^+ is the domain of positive integers. The *reply* function is then defined for each distinct state of each protocol, identified by a unique (for each protocol) integer. This gives for each speech act, 'performed' in the context of a conversation being conducted according to a specific protocol, what performatives in which protocols are acceptable replies. The *reply* function therefore specifies a finite state diagram for each protocol named in Prot.

To fully characterise the intended semantics, three further functions are required, which are specified relative to each agent a, and state what that agent does with a message, not how it does it. The three functions in [12] were (1) a procedure for computing the change in an agent's information state from the content of an incoming message; (2) a procedure for selecting a performative from a set of performatives (valid replies), and (3) a function *conv* which mapped a conversation identifier onto the current state of the protocol. For these functions, we specify intentional (logical) descriptions of the reasons for and reactions to a speech act. These serve as reference implementation models that agent developers could use to implement the appropriate internal and external behaviours for their agents. Furthermore, where the import of the the content level meaning was required, further specifications could be supplied, and this is dependent upon the application.

An agent s then communicates with (and communicates information to) an agent r via a speech act. This (possibly infinite) set is denoted by *speech_acts*, a single member sa of which is represented by:

$$sa = \ll s, \mathsf{perf}(r, C, L, O, cp, ci, t_s) \gg$$

This is saying that s does (communicates with) performative perf with content C in language L using ontology O in the context of protocol (conversation policy) cp as part of a conversation identified by ci at time of sending t_s. The notation $sa.perf$ denotes the performative of a speech act, and so on.

The meaning of such a speech act sa from agent s to agent r is then given by:

$$[\![\ll s, \mathsf{perf}(r, (C, L, O, cp, ci, t_s)) \gg]\!] = \mathcal{I}_r \ll r, rep\_sa \gg \text{ s.t.}$$
$$(rep\_sa.perf, rep\_sa.prot) \in reply(\mathsf{perf}, cp, conv_r(ci))$$

This means that, in this framework, at the observable action level, the meaning of a speech act is the intention to give a reply.

We next show how extend this framework based on a semantics for protocol diagrams. In section 5, we associate the threads of interaction with intentional specifications, and integrate observable commitments with the interaction patterns and these specifications.

4.2 Semantics of Protocol Diagrams

The usefulness of finite state diagrams for protocol representation was that they could be given a mathematical formulation via the *reply* function. At each state (stage of the conversation between two agents), an agent had a set of possible replies. The intentional specifications decided which of these was actually performed, and this of course was consistent with the external meaning of a speech act.

We attempt here to give a similar formal meaning to a UAML protocol diagram. In particular, we use the threads of interaction in the same way that states were used in finite state diagrams.

We stipulate that all threads of interaction are either:

- a unique starting thread;
- start with receipt of a message, end with despatch of a message;
- start with receipt of a message and terminate, ending the protocol.

We then claim that a protocol diagram consists of:

- a finite number of distinct threads per role; and
- a finite number of sequences of messages (communicative acts) from the start thread to a terminal thread.

We can then use the sequence of communicative acts to identify the thread of interaction an agent is 'in'. The thread determines the space of possible replies.

Let *Perf* be the set of communicative acts in an ACL, and *Prot* the set of protocols, as before. Then let *Rol* be the set of possible agent roles, and *Thrd* an infinite set of thread identifiers $t_1, t_2, \ldots, t_n, \ldots$. Let Σ be the domain of sequences of speech acts, i.e. for each $\sigma \in \Sigma$:

$$\begin{aligned}\sigma = \; < \; & \ll sndr, perf(rcvr, \ldots, prot, \ldots) \gg, \\ & \ldots, \\ & \ll rcvr, perf(sndr, \ldots, prot, \ldots) \gg \; >\end{aligned}$$

Then define functions:

$$reply_1 : Perf \times Prot \times Rol \times \Sigma \longrightarrow Thrd$$

and:

$$reply_2 : Prot \longrightarrow Thrd \longrightarrow \mathbf{P}(Perf \times Prot)$$

So, for example, for our contract net protocol (*cnp*), we have a $reply_1$ function which returns values such as[2]:

$$\begin{aligned}reply_1(cfp, cnp, e, <>) &= t_5 \\ reply_1(reject, cnp, c, < cfp >) &= t_1 \\ reply_1(accept, cnp, c, < cfp >) &= t_2, \text{if before timeout} \\ &= t_2, \text{if after timeout}\end{aligned}$$

[2]We only show the performative names in each sequence.

Note from the final value, that the value returned can take into account the deadlines, roles, cardinalities, sequences of messages, and so on. These specifications can be arbitrarily complex, and intuitively should be able account for anything that can be graphically represented on a protocol diagram.

Given the type of the $reply_2$ function as $\lambda P \lambda t.P(t)$, we also have:

$$\begin{array}{rcl} cfp(t_5) & = & \{(propose, cnp), (reject, cnp)\} \\ cfp(t_1) & = & \{\} \\ cfp(t_2) & = & \{(accept, cnp), (reject, cnp)\} \end{array}$$

Having the domain of $reply_2$ as the powerset of performatives and protocols allows new protocols to be started [11], or nested protocols [6]. For full generality, this domain should probably be sequences of elements from this domain, to allow for multiple communicative acts to be performed from a single thread of interaction.

From these functions, and given (1) a function (essentially a dynamic loop-up table) that maps an agent name to its role in a conversation (conducted according to the protocol), and (2) an adjustment to the *conv* function that returns sequences of communicative acts rather than states, i.e.

$$conv_a : Cid \longrightarrow \Sigma$$

we can define the meaning of a speech act exactly as before, i.e. as an intention to reply with allowed message type:

$$reply_2(cp)(reply_1(perf, cp, role(r, ci), conv_r(ci)))$$

We have then given a semantics to UAML diagrams and integrated this semantics with our general semantic framework for ACLs. We now proceed to give a full axiomatisation of the contract net protocol.

5 Intentional Specifications

The purpose of intentional specifications is to give a reference implementation model of the decision-making in the context of a protocol. This specifies which of the possible replies should be made, and we make reference to the beliefs, desires and intentions of the agent (although this is not mandatory for implementation).

The intentional specifications are used to state what we call triggers and tropisms [10]. The idea is that triggers are the combination of beliefs and desires that produce intentions (to do actions), and that tropisms are the affects on beliefs and desires that results from executing those intentions.

Notice that triggers and tropisms specified like this are local to each individual agent, and while they may be identical between a sender and receiver agent in a closed multi-agent system, it is possible for them not to be (for example in an open system with agents deployed by different organizations). Therefore, we have the following categorization:

	trigger	tropism
sender's side	sender's actual motivation for doing an action	sender's belief about expected outcome of action on receiver's belief state
receiver's side	receiver's belief about motivation (sender's belief state) for doing an action	actual result (receiver's actual change of belief state) of doing action

In this section, we give intentional specifications, in the form of triggers and tropisms, for the contract-net protocol given above. We will then augment this specification with the outcomes, norms, and obligations of the protocol. These, we argue, should be part of the protocol diagram specification as 'externals' or observable commitments, which we hook into our intentional specifications.

5.1 Intentional Specification Language

We follow many existing works which use beliefs, desires and intentions (BDI) to characterise the semantics of performatives [14, 3]. The formal syntax of our language for writing trigger and tropisms is a first-order modal logic with relativised belief, desire, and intention (to) modalities $\mathcal{B}_a$, $\mathcal{D}_a$ and $\mathcal{I}_a$ respectively; action formulas written $\ll a, A \gg$ and a DONE operator on action formulas; and parameterised action modalities $[a, A]$ (read as "after agent a does action A"). Once an intention to do an action is executed then DONE of the action is true: i.e. given $\mathcal{I}_a \ll a, A \gg$, as an intention, then after execution $\text{DONE}(a, A)$ is true. Furthermore, if an agent did an action, then the action was done, i.e. $\text{DONE}(a, A) \rightarrow \text{DONE}(A)$; and if an agent did an action, then we can infer that the agent *brought it about that* or *saw to it that* the action was done, i.e.:

$$\frac{\text{DONE}(a, A)}{\mathcal{E}_a \text{DONE}(A)}$$

The relativised modality $\mathcal{E}_x$ used here is the Jones and Sergot [9] action modality used for expressing the idea that agent x creates or establishes a state of affairs, and that x performs designated acts. See [9] for a formal characterisation of this modality.

To give a formal specification of intentional behaviour, we want to write axioms of the form (for any agent a):

$$\models \mathcal{B}_a \phi \wedge \mathcal{D}_a \psi \rightarrow \mathcal{I}_a \ll a, A \gg$$
$$\models [a, A]\phi \rightarrow \chi$$

Here, $\mathcal{B}$, $\mathcal{D}$ and $\mathcal{I}$ are relativised (agent) modalities for beliefs, desires and intentions, and $[a, A]$ is an agent-action modality. The intuitive reading of these axioms is then firstly, that if an agent believes ϕ and desires ψ, then it will form the intention to perform action A. Secondly, that after agent a performs action A, χ holds. [10] discusses a semantics for this language and its relation to the BDI achitecture.

5.2 Preliminaries

We need the following notation for multi-cast, predicates that hold over groups, and state variables.

A multi-cast speech act by an agent s to a set of known receivers R is written[3]:

$$\ll s, \mathit{multicast}(\mathsf{perf}, R, C, P) \gg \quad \stackrel{def}{=} \mathop{;}_{r \in R} \ll s, \mathsf{perf}(r, C, P) \gg$$

Here, ';' is the conventional operator for a sequence of actions. This notation then describes the multi-cast of a performative perf, to a set of receivers R, with content C, in protocol P, performed by an agent s as the composition of a sequence of individual acts to each $r \in R$.

A predicate holding over a group (i.e. a set of agents) is defined as:

$$p_G(S) \quad \stackrel{def}{=} \quad \bigwedge_{a \in S} p(a)$$

We also need to refer to *state variables* [7, 11], which are global identifiers whose values are changed by communicative acts. In this case, we need three: B, the set of bidders who respond to a *cfp* with an *accept* (before the deadline), initially empty; w, the winning agent selected from this set by the contractor, whose initial value is undefined; and $timer$, whose value can be agreed when the contract net is set up or included as part of the content of the *cfp*.

Access and change to a state variable $svar$ is indicated by the following notation:

$conv_a(i).svar = val$	to test the value for equality
$conv_a(i).svar \leftharpoondown newval$	to overwrite the current value

5.3 Contract-Net: Triggers and Tropisms

The trigger for initiating a contract-net protocol to a set of potential bidders is:

$$\begin{array}{l} \mathcal{D}_c \mathrm{DONE}(T) \wedge \mathcal{B}_c capable_G(B, T) \rightarrow \\ \qquad \mathcal{I}_c \ll multicast(\mathsf{cfp}, B, T, cnp) \gg \end{array} \tag{1}$$

This states that if a contracting agent c desires a task T to be performed and believes that each member b of a set B is capable of performing that task, then it will start a contract-net protocol by multi-casting a cfp to each agent in B.

The tropisms for the multi-party contract-net protocol are formally specified (and paraphrased) by the intentional specifications shown below, for the communicative acts received and the thread of interaction tx (see figure 3) this causes. Note the only parameters we are using are the receiver, the task being proposed T, and the protocol (cnp, i.e. the contract net protocol). However, following [11] it is understood that a conversation between multiple-agents is marked by a unique conversation identifier cid, which is a parameter to all messages. The *timeout* action (an event that occurs during thread $t2$) then refers explicitly to this conversation. Furthermore, all reference to state variables S, are technically $conv_c(cid)S$, etc., but we simplify for clarity.

[3] We omit the parameters for ontology, time and conversation identifier for clarity. For full details, see [11].

$$
\begin{array}{lll}
t4 & [c, \mathsf{cfp}(e, T, cnp)] & (2) \\
 & \quad \mathcal{B}_e bid(T) \rightarrow \mathcal{I}_e \ll e, \mathsf{propose}(c, T, cnp) \gg & \\
 & \lor \quad \neg \mathcal{B}_e bid(T) \rightarrow \mathcal{I}_e \ll e, \mathsf{reject}(c, T, cnp) \gg & \\
 & [e, \mathsf{propose}(c, T, cnp)] & (3) \\
t2 & \quad timer > 0 \rightarrow S \leftharpoondown S \cup \{e\} & \\
t1 & \lor \quad timer \leq 0 \rightarrow \mathcal{I}_c \ll c, \mathsf{reject}(e, T, cnp) \gg & \\
t2 & [c, timeout(cid)] & (4) \\
 & \quad select\_winner(S, w) \land & \\
 & \quad S \leftharpoondown S \backslash \{w\} \land & \\
 & \quad \mathcal{I}_c \ll e, \mathsf{accept}(w, T, cnp) \gg \land & \\
 & \quad \mathcal{I}_c multicast(\mathsf{refect}, S, T, cnp)) & \\
t5 & [c, \mathsf{accept}(w, T, cnp)] & (5) \\
 & \quad \mathcal{D}_w \mathrm{DONE}(T) \land & \\
 & \quad \mathcal{D}_w \mathrm{DONE}(pay(w, T)) & \\
t3 & [w, \mathsf{inform}(c, \mathrm{DONE}(T), cnp)] & (6) \\
 & \quad \mathcal{B}_c \mathrm{DONE}(w, T) &
\end{array}
$$

When the contractor issues it's accept, this we argue, is the point when the *contract* (i.e. the point of a contract-net protocol) between the contractor and the bidder is established. The action of agent c then *counts as* a means of creating another state of affairs. This state of affairs is sanctioned by some institution which will enforce the contract.

This idea of 'counting as' can be formalised using the Jones and Sergot [9] conditional operator $\stackrel{s}{\Rightarrow}$. The reading of formulas like:

$$\mathcal{E}_a F \stackrel{s}{\Rightarrow} \mathcal{E}_b P$$

is that if, on some occasion, agent s sees to it that F holds, then within the institution s the state of affairs F holds. F is then a matter of fact, relative to institution s, i.e. what Searle referred to an institutional fact [15]. For full formal details of this new logical operator, see [9].

If an agent a brings it about that an accept is performed in the contract-net protocol, then this counts as establishing a contract – according to the institution S – between the sender and receiver to do the task T, . Formally, we have:

$$\mathcal{E}_c \mathrm{DONE}(\mathsf{accept}(w, T, cnp)) \stackrel{s}{\Rightarrow} \mathcal{E}_S contract(c, w, T) \qquad (7)$$

this notion of contract, we specify the following trigger axioms. First, for the winning bidder, we have:

$$\mathcal{B}_w \mathcal{E}_S contract(c, w, T) \land \mathcal{D}_w \mathrm{DONE}(T) \rightarrow \mathcal{I}_w \ll w, T \gg (8)$$

$$
\begin{array}{l}
\mathcal{B}_w \mathcal{E}_S contract(c, w, T) \land \\
\quad \mathcal{B}_w \mathrm{DONE}(T) \land \mathcal{D}_w \mathrm{DONE}(c, pay(w, T)) \rightarrow \\
\qquad \mathcal{I}_w \ll w, \mathsf{inform}(c, \mathrm{DONE}(T), cnp) \gg \qquad (9)
\end{array}
$$

and for the contractor we have:

$$
\begin{array}{l}
\mathcal{B}_c \mathcal{E}_S contract(c, w, T) \land \mathcal{B}_c \mathrm{DONE}(w, T) \rightarrow \\
\qquad \mathcal{I}_c \ll c, pay(w, T) \gg \quad (10)
\end{array}
$$

Note that the receipt of the final inform causes the contractor to believe that task T has been done by w. This discharges the original desire that initiated the contract-net protocol. It also creates the additional intention to pay w, the execution of which will discharge w's outstanding desire, which is to be paid for doing the task.

5.4 Observable Commitments

We have now seen how a protocol can create a contract, or more generally a commitment to a state of affairs. Furthermore, there are additional constraints on the agents' behaviour, which are the conditions that the agents should observe when using the speech acts (in the context of this protocol). These are the *norms* [9] of the agent society in which this protocol is used.

For example, we can state (in English) that the norms to be observed by the agents in the contract net protocol are:

- if a contractor puts a task out to tender, it should be able to pay for it (committed to paying for it);
- if a bidder offers to perform a task, it should be able to do it (committed to doing it);
- at the end of the protocol, the winner is obliged to perform the task for the contractor;
- after the task has been performed, the contractor is obliged to pay the winner the agreed amount.

Therefore, we extend the logical language with the following notation (cf. [4]):

$\mathcal{C}_a^s(F;P)$	it is a norm of society s that when F is true, then a is committed to P being true;
$\mathcal{O}_a^s(F;P)$	in society s, when F is true, it is a obligation of a to see to it that P be true.

We can then formalise the four norms above as follows:

$$\mathcal{C}_c^S(\mathrm{DONE}(\ll c, \mathsf{cfp}(e,T,cnp) \gg); \mathrm{DONE}(pay(T))) \quad (11)$$
$$\mathcal{C}_e^S(\mathrm{DONE}(\ll e, \mathsf{offer}(c,T,cnp) \gg); \mathrm{DONE}(T)) \quad (12)$$
$$\mathcal{O}_w^S(contract(c,w,T); \mathrm{DONE}(w,T)) \quad (13)$$
$$\mathrm{DONE}(w,T) \rightarrow \mathcal{O}_c^S(contract(c,w,T); pay(w,T)) \quad (14)$$

For formal completeness, each type of norm $\mathcal{C}$ and $\mathcal{O}$ above is an instance of what Jones and Sergot [9] refer to as an institutional constraint. These are a general way of characterising the various different conditions on an institution, of which 'counts as' is one sort. [9] propose a relativised normal modality $\mathcal{D}_S$ as a general notion, with axioms to define the relations to specific cases such as 'counts as'.

We can therefore define the norms above as general institutional constraints (with *Can* a practical ability operator and **O** deontic obligation [9]) as follows:

$$\mathcal{C}_a^S(F;P) \stackrel{def}{=} \mathcal{D}_S(F \rightarrow Can\mathcal{E}_a P)$$
$$\mathcal{O}_a^S(F;P) \stackrel{def}{=} \mathcal{E}_S F \rightarrow \mathcal{D}_S \mathbf{O} \mathcal{E}_a P$$

This means that a commitment, so far as society S is concerned, is that (if F) a has the practical ability to see to it that P; and a normative obligations is that if the society sees to that F, then it is a constraint of the society that a is obliged (in the deontic sense) to see to it that P.

6 Animation

Two or more agents communicating with each other, using this protocol, should respect the specified external semantics, irrespective of how they are actually implemented. This means that they should make expected replies in response to received speech acts, and also should respect and comply with the norms, obligations and commitments annotating the protocol (which are also external).

To demonstrate this compliance, we animate the specification above. The logical animation in Table 1 shows the sequence of events and changes to the belief, desires and intentions of two agents, a contractor c and a bidder e. For simplicity we consider only one bidder and so this will be the 'winner'; it should be clear how the animation scales up in the presence of multiple bidders.

We now show how this sequence of events complies with the four norms of section 5.4. Notice the following derivation:

$$
\begin{array}{lll}
 & \mathcal{I}_c \ll c, \mathsf{accept}(e, T, cnp) \gg & \\
\Rightarrow & \text{DONE}(c, \mathsf{accept}(e, T, cnp) \gg & \\
\Rightarrow & \mathcal{E}_c\text{DONE}(\mathsf{accept}(e, T, cnp)) \gg & \\
\Rightarrow & \mathcal{E}_S\, contract(c, e, T) & \text{by (7)} \\
\Rightarrow & \mathcal{D}_S \mathbf{O} \mathcal{E}_e \text{DONE}(e, T) & \text{by (13), defn. of } \mathcal{O}
\end{array}
$$

The point here is that the exchange of messages in the protocol creates the obligation on the winner as an institutional constraint. This obligation is to see to it that task T is done, which checking the animation above, it does. Thus e respects the norm (13). A similar derivation leads to the obligation on c via norm (14), and in the final line of the animation, c respects this obligation too.

In addition, the specification or implementation of *bid* should evaluate to true only if the bidder is capable of performing the task, in order to comply with norm (12). Finally, the trigger for the contract-net protocol should have an extra condition $can\_pay(c, T)$ in order to be sure that a contracting agent will comply with norm (11).

7 Summary and Conclusions

7.1 The Communicative Context

We started out by considering a contract-net protocol for the producer-consumer scenario outlined in section 2. We used AUML for specifiying the protocol, but were more concerned with analysing the interactions from a 'social' perspective, focussing on norms (right, permissions, obligations etc.). The results have been: a proposed semantics for AUML diagrams, integration of this proposal with our general semantic framework for ACLs, enhancement of AUML diagrams with observable commitments (norms and outcomes), and integration of these within the same general framework.

The significance of this is:

- our proposed methodology for designing ACLs starts with AUML protocol diagrams, a richer and more expressive notation than before, but integrated with the method;
- we have an extra level of *external* behaviour for checking compliance to a standard (respecting the norms);

Table 1: Logical Animation of Contract-Net Specification

Assume, at time t

$\mathcal{B}_c capable(e, T)$

$\mathcal{D}_c \text{DONE}(T)$

intention generated, by trigger (1) and defn. of multicast

$\mathcal{I}_c \ll c, \mathsf{cfp}(e, T, cnp) \gg$

intention executed, message sent, state variables initialized

$S \leftharpoondown \emptyset$

$w \leftharpoondown \bot$

$timer \leftharpoondown 50$

message received, tropism (2) active

$bid(T)$ evaluates to true (say)

intention generated

$\mathcal{I}_e \ll e, \mathsf{propose}(c, T, cnp) \gg$

intention executed, message sent

message received, within timeout, so by tropism (3)

$S \leftharpoondown S \cup \{e\}$

...

timeout, so by tropism (4)

$select\_winner(S, w)$ evaluates to true with $w == e$

intention generated

$\mathcal{I}_c \ll c, \mathsf{accept}(e, T, cnp \gg$

intention executed, message sent, so by counts as (7)

$\mathcal{B}_c \mathcal{E}_S contract(c, w, T)$

message received, by tropism (5)

$\mathcal{D}_e \text{DONE}(T)$

$\mathcal{D}_e \text{DONE}(c, pay(w, T)$

and by counts as (7)

$\mathcal{B}_e \mathcal{E}_S contract(c, w, T)$

intention generated, by trigger (8)

$\mathcal{I}_e \ll e, T \gg$

intention executed

$\mathcal{B}_e \text{DONE}(e, T)$

desire of e for DONE(T) *discharged*

intention generated, by trigger (9)

$\mathcal{I}_e \ll e, \mathsf{inform}(c, \text{DONE}(T), cnp) \gg$

intention executed message sent

message received, by tropism (6)

$\mathcal{B}_c \text{DONE}(e, T)$

desire of e for DONE(T) *discharged*

intention generated, by trigger (10)

$\mathcal{I}_c \ll c, pay(e, T) \gg$

intention executed, payment received by e

$\mathcal{B}_e \text{DONE}(c, pay(e, T))$

remaining desire of e discharged.

- we have established a causal connection between what agents say, what they do, and what they are committed to.

In previous work, we have been concerned with defining, through the intentional specifications, what is in 'the mind of an agent' when it decides to communicate. The next issue to consider is: who is responsible for it. The EU-sponsored ALFEBIITE project[4] is investigating a legal framework for normative behaviour of and between autonomous agents. In particular we are concerned with the implications of agent communication, and the 'social' commitments that arise as a result. The current work is a contribution to this research programme, and refines the general communicative context for agent interaction (as illustrated in figure 4).

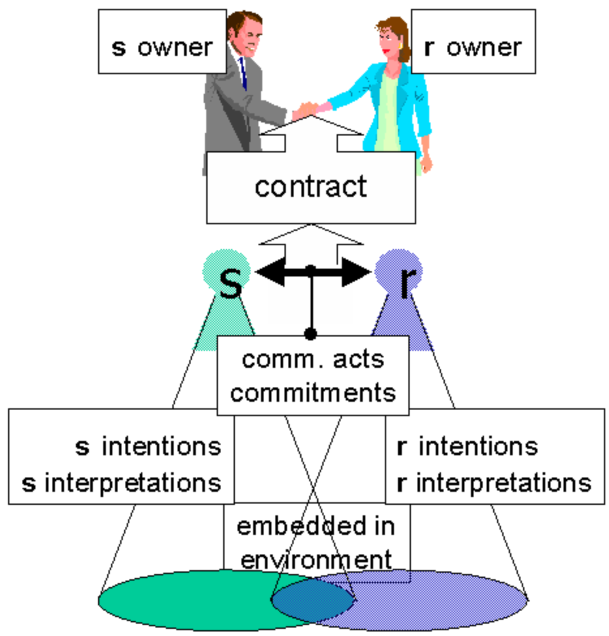

Figure 4: The Communicative Context for Agents

We had previously been concerned with three levels of meaning: the action level (speech acts and replies), the intentional level (the 'mind of an agent'), and the content level (interpretation). We argues that the first was external to the agents (and therefore amenable to standardization), while the latter two were not. We are now defining meaning at a *fourth* level, what we might call the *social* level, and this too is external. As such, compliance should also be verifiable, as was demonstrated by the animation in section 6. Furthermore, based on this external semantics, contracts (or other social relations) between agents can have concrete counterparts between human entities (individuals or organizations) in the 'real' world. This is ultimately the relationship we want to capture in ALFEBIITE, and goes some way to characterising responsibility and liability for autonomous agent behaviour.

[4]http://www.alfebiite.com/

7.2 Parameterisation of Protocols

The state variables identified in Section 5.2 are part of the contract net protocol specification. Therefore, they should reasonably be specified in the protocol diagram, and this can be conveniently done by detailing all such variables in an annotation to the diagram (as will parameters). In addition, the two previous sections have shown that the outcome of the protocol – what the agents are negotiating about – and the constraints on their behaviours in achieving this outcome, should also be specified.

We therefore propose that the AUML protocol diagrams should represent state variables, observed norms during the protocol, and outcomes from successfully concluding a negotiation according to the protocol. We would suggest that AUML protocol diagrams should appear as illustrated in Figure 5.

$$\mathcal{E}_c\text{DONE}(accept(w, T, cnp)) \overset{s}{\Rightarrow} \mathcal{E}_S contract(c, w, T)$$

$$\mathcal{N}^S(\text{DONE}(\ll c, \mathsf{cfp}(e, T, cnp) \gg); can\_pay(c, T))$$
$$\mathcal{N}^S(\text{DONE}(\ll e, \mathsf{offer}(c, T, cnp) \gg); capable(e, T))$$
$$\mathcal{O}_w^S(contract(c, w, T); \text{DONE}(w, T))$$
$$\text{DONE}(w, T) \rightarrow \mathcal{O}_c^S(contract(c, w, T); pay(w, T))$$

$$S \leftharpoondown \emptyset$$
$$w \leftharpoondown \bot$$
$$timer \leftharpoondown 50$$

Figure 5: Parameterised AUML Protocol Diagram

Speech acts in a protocol should respect the norms of that protocol, in addition to the semantics specified by the ACL. This is because speech acts *count as* a certain type of utterance, with associated conditional effects, as a result [9]. [1] insist that the semantics of the messages in a protocol are consistent with the semantics of the individual speech acts, as for example, documented in [5] for the FIPA ACL.

We have argued elsewhere that this is too restrictive [13]. The virtue of the current proposal it that it makes *all* such specifications a a norm and a parameter of the protocol. In this way, by specifying a norm we identify a property that should be observed and may be punished by transgression, and as a parameter it is a condition that can be dropped when it is not mandated.

For example, agents utterly faithful to the FIPA-ACL semantics should only send content which they believe to be true. However, this is, in fact, a norm of a society S using FIPA-ACL as its content language:

$$\mathcal{N}^S(\text{DONE}(\ll s, \mathsf{inform}(r, p, \ldots) \gg); \mathcal{B}_s p)$$

This is then a norm which should be observed, but allows for non-normative behaviour (and punishment for such), but as a parameter can be dropped if an application demands that so-called self-interested agents are allowed to relax the 'sincerity condition' for their purposes.

7.3 Evaluation of AUML

The move to a protocol-oriented semantics for FIPA-ACL is, in our opinion, a positive one, and the proposal to embrace the highly-successful modelling technique of UML is significant, in order to 'mainstream' agent-based software engineering and FIPA standards.

However, our experience of using AUML to specify the contract-net protocol in AUML was not without difficulties. For us, this included:

- limits of representation: the examples we have seen have tended to be small and effectively have one successful sequence of messages, and few, quickly terminated 'branches'. Protocols which have many x-or decisions lead to extremely cluttered diagrams;
- representation of time and exogeneous events: the agent lifeline implies a 'flow of time' down the diagram, and the notation supports a 'time intensive' message (denoted by a downward sloping arrow). However, the length of a thread is more often for presentation purposes and not indicative of time, and [1] have a message arrow going 'backwards';
- representation of errors: there is little support for undelivered messages or communication errors, and care has to taken to avoid protocol 'explosion'. For example, the FIPA specification has FIPA-Request and FIPA-Request-when-cancellable protocols, and similarly for contract net. This can be expanded almost indefinitely. A separation of concerns is required;
- a semantics is required. The current FIPA document [6] seems to favour form over content ("A small dashed rectangle is superimposed on the upper right-hand corner of the rectangle with rounded corners like defining a nested protocol." is not atypical). The semantics proposed here is a contribution to that effort;
- focus on essential information: allowing parameterized protocols is good, but the essential information needs to be included. The proposal above, focusing on norms and outcomes, is strongly proposed for inclusion in the standard.

7.4 Further Work

We conclude this paper we some items of further work that build on the results achieved:

- more protocols are being defined for the scenario; of particular interest are those protocols whose outcomes are other types of norm, for example permissions, powers, etc.;
- a general agent inference engine is being implemented, which can reason directly with the intentional specifications, underlying a general-purpose BDI architecture;
- the scenario and the agent engine facilitates experimentation with formal models of agent societies, and other social relations like delegation, trust and control (see, e.g., [2]);
- a more thorough investigation of the relationship between the content of the contract-net, abstractly specified as T above, and service level agreements [8];
- generalizing this work within a unified account of agent communication languages.

Further theoretical development of this work along these lines, together with continuing implementation and experimentation, are the main focus of our current investigations.

Acknowledgments

This work has been undertaken with the financial support of the EU-funded project ALFEBIITE (IST-1999-10298). The interaction with partners in this project, in particular Andrew Jones and Marek Sergot, has been of particular benefit to the current work.

References

[1] B. Bauer, J. Muller, and J. Odell. An extension of UML by protocols for multiagent interaction. In *Proceedings ICMAS-2000*, pages 207–214. IEEE Computer Society, 2000.

[2] C. Castelfranchi and R. Falcone. Towards a theory of delegation for agent-based systems. *Robotics and Autonomous Systems*, 24:141–157, 1998.

[3] P. Cohen and H. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42(2-3):213–261, 1990.

[4] F. Dignum, D. Morley, E. Sonenberg, and L. Cavedon. Towards socially sophisticated bdi agents. In *Proceedings ICMAS-2000*, pages 111–118. IEEE Computer Society, 2000.

[5] FIPA. Communicative act library specification. FIPA (Foundation for Intelligent Physical Agents), Document XC00037, http://www.fipa.org, 1999.

[6] FIPA. Interaction protocol library specification. FIPA (Foundation for Intelligent Physical Agents), Document XC00025, http://www.fipa.org, 2000.

[7] F. Guerin and J. Pitt. A semantic framework for specifying agent communication languages. In *Proceedings ICMAS-2000*, pages 395–396. IEEE Computer Society Press, 2000.

[8] N. Jennings, P. Faratin, T. Norman, P. O'Brien, B. Odgers, and J. Alty. Autonomous agents for business process management. *International Journal of Applied Artificial Intelligence*, 14(2):145–189, 1996.

[9] A. Jones and M. Sergot. A formal characterisation of institutionalized power. *Journal of the Interest Group in Pure and Applied Logics*, 4(3):429–455, 1996.

[10] J. Pitt. An operational semantics for intentional specifications of agent behaviour in communication protocols. In M. Greaves, editor, *Proceedings AA2000 Workshop on Agent Languages and Conversation Policies*, pages pp31–40. Barcelona, 1999.

[11] J. Pitt, F. Guerin, and C. Stergiou. Protocols and intentional specifications of multiparty agent conversations for brokerage and auctions. In *Proceedings Autonomous Agents 2000*, pages 269–276. ACM Press, 2000.

[12] J. Pitt and A. Mamdani. A protocol-based semantics for an agent communication language. In *Proceedings 16th International Joint Conference on Artificial Intelligence IJCAI'99*, pages 485–491. Morgan-Kaufmann, 1999.

[13] J. Pitt and A. Mamdani. Some remarks on the semantics of FIPA's agent communication language. *Autonomous Agents and Multi-Agent Systems*, 2(4):333–357, 1999.

[14] D. Sadek. A study in the logic of intention. In *Proceedings 3rd Conference on Principles of Knowledge Representation and Reasoning (KR'92)*, pages 462–473. Cambridge, MA, 1992.

[15] J. Searle. *Speech Acts: An Essay in the Philosphy of Language.* Cambridge University Press, 1969.

[16] M. Witkowski and J. Pitt. Objective trust-based agents: Trust and trustworthiness in a multi-agent trading society. In *Proceedings ICMAS-2000*, pages 463–464. IEEE Computer Society, 2000.