A Protocol-Based Semantics for an Agent Communication Language

Jeremy Pitt and Abe Mamdani
Intelligent & Interactive Systems Group, Dept. of Electrical & Electronic Engineering
Imperial College of Science Technology and Medicine, London SW7 2BT, England
Email: {j.pitt,aanamdanij@ic.ac.uk URL: http://www-ics.ee.ic.ac.uk

Abstract

There are fundamental limitations on using mental attitudes to formalise the semantics of an Agent Communication Language (ACL). Instead, we define a general semantic framework for an ACL in terms of protocols. We then argue that the proper role of mental attitudes is to link what an agent 'thinks' about the content of a message to what it 'does' in response to receiving that message. We formalise this connection through normative and informative specifications and demonstrate its use in communication between two BDI-style agents.

1 Introduction

The growing ease of network connectivity of computers provided the enabling technology for the agent paradigm. From the computing perspective, agents are autonomous, asynchronous, communicative, distributed and possibly mobile processes. From the AI perspective, they are communicative, intelligent, rational, and possibly intentional entities.

The common feature of communication has determined that some kind of message passing between agents is required. To provide inter-operability between heterogeneous agents, a commonly understood agent communication language (ACL) is used: examples include KQML [Finin *et* al., 1995], Arcol [Breiter and Sadek, 1996], and FIPA's ACL [FIPA, 1997]. To ensure that it is commonly understood, a formal semantics for the ACL is required. From the AI perspective, the semantics has typically been characterised in terms of speech act theory and framed in terms of the intentional stance, i.e. mentalistic notions such as beliefs, desires and intentions [Cohen and Levesque, 1995; Breiter and Sadek, 1996; FIPA, 1997].

An intentional normalisation of an ACL semantics often involves the axiomatisation of the felicity conditions of Searle and the conversational maxims of Grice. Grice's analysis of conversational irnplicatures was underpinned by the sincerity condition to support the co-operativity principle. For agents to converse co-operatively (e.g. for negotiation, co-ordination, etc.), the sincerity condition is a reasonable requirement, but is often hard-wired into the semantics as a feasibility precondition on performing a simple speech act, e.g. KQML tell and FIPA inform. Thus an agent believes what it says and only says what it believes.

However, by defining the meaning of a performative in isolation through an axiomatisation that implicitly expects the performative to be used in conversation, there is a risk of being exclusive. In isolation, we could motivate the behaviour of a sincere agent by the axiom:

$$\models \mathcal{B}_s\phi \wedge \mathcal{D}_s\mathcal{B}_r\phi \rightarrow \mathcal{I}_s(\text{DONE}(< s, inform(r, \phi) >))$$

Here $\mathcal{B}$, $\mathcal{D}$ and $\mathcal{I}$ are respectively the beliefs, desires (goals) and intends modalities, and DONE is an operator on actions. This axiom therefore states that if the agent *s* believed ϕ and wanted another agent r to believe ϕ, then 8 would generate the intention to inform r of ϕ, after which action r may also come to believe ϕ. Alternatively we could define the behaviour of a 'rapacious' agent by the axiom:

$$\models \mathcal{D}_s\psi \wedge \mathcal{B}_s(\text{DONE}(A) \rightarrow \psi) \rightarrow \mathcal{I}_s\text{DONE}(A)$$

This axiom states that if .s desires (wants) ψ and believes that doing *A* will achieve ψ then s will intend to do *A*. If $\psi = \mathcal{B}_r\phi$, the same communication occurs (A is $< s$, *inform* $(r, \phi) >$), without s having an explicitly held belief in ϕ.

Any formalisation of communication based on the mental states of the participants must deal with what is fundamentally a process of revision and updating of those states. It is unlikely that a single set of axioms will cover all eventualities because communication is inherently context-dependent. What works in one application may be inappropriate in another. Furthermore, intentionality is concerned with agent internals and a communicative act is an external phenomenon: mental attitudes can only give a possible reason for and not the definitive meaning of the performative. Therefore there are considerable limitations of intentionality as a basis for defining the semantics of an ACL (cf. [Singh, 1998]).

This paper develops an alternative semantic framework for an ACL from the computing perspective, with an emphasis on protocols. Mental attitudes are used now to link what an agent 'thinks' about the message content to what it 'does' in response to receiving that message.

2 A Layered Semantics

Consider the situation illustrated in Figure 1. It shows two agents, each embedded in an environment, which partially overlap. However, rather than communicating by changing the environments, they can communicate by using speech acts and the ACL protocols.

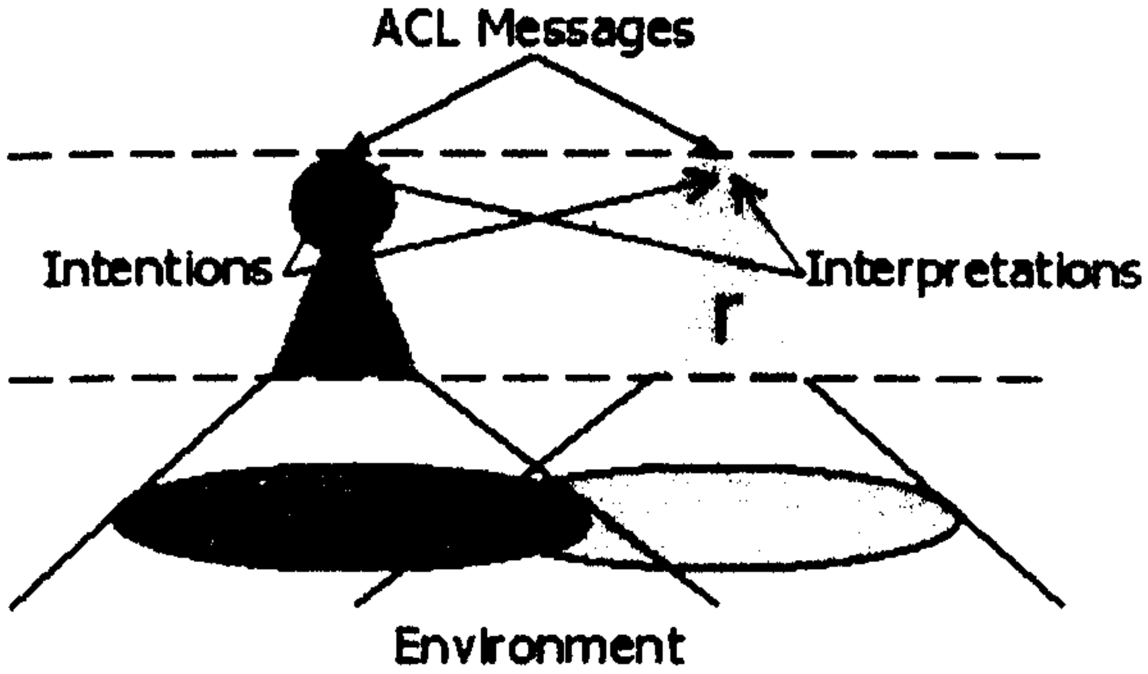

Figure 1: Communicating Agents

We identify three layers of semantics here:

i The content level semantics, which is concerned with interpreting and understanding the content of a message, and is internal to an agent;
ii The action level semantics, which is concerned with replying in appropriate ways to received messages, and is external to the agents;
iii The intentional semantics, which is concerned with making a communication in the first place, and with replying, and again is internal to the agent.

We would argue that the current FIPA ACL semantics for example, is level 3, and because it is internal to an agent, its usefulness in standardisation lias been questioned [Wooldridge, 1998; Singh, 1998]. The only part of the communication that is amenable to standardisation is the observable tip of the iceberg: namely the communication itself. Note that this properly includes ontologies, so that there may be a standard interpretation, but the actual interpretation of the content of the message is once again internal to the agent.

3 Generic Semantics for Performatives

We define a standard semantics for performatives by an input-output relationship at the action level (level (ii) above). We define the meaning of a speech act (as input) as the intention to perform another speech act (as output). In computing this functional relationship we can take into account the agent's mental state, without proscribing that state or how it should be implemented.

We base the relationship on how the object-level content of a message induces a change in the information state of the receiver, and how the meta-level action descriptor of a message (the performative itself) induces a response from the receiver. Our proposal is that the semantics of performatives can be characterised in these terms, and that this is the semantics that should be specified for a standard agent communication language. This means specifying, for the content, what it 'means' for an agent (receiving a message) to add its interpretation of that content to its current information state; and for the speech act, the space of possible responses an agent may make, one of which it is committed to make.

For the specification that follows in this section, let *c* and *a* be sets of integers, and let Δ, *content.meanings,* and *speech-acts* be, respectively, possibly infinite sets of agent information states, the agent's interpretation of the content of a message, and speech acts.

An ACL is defined by a 3-tuple <Perf, Prot, *reply*> where Perf is a set of perfomative names, Prot is a set of protocol names, and *reply* is a partial function given by:

$$reply : \text{Perf} \times \text{Prot} \times \sigma \mapsto \wp(\text{Perf})$$

For the FIPA ACL, for example, Perf would be the set of all FIPA ACL performatives, and Prot would be the set of all FIPA ACL protocols, used for negotiations, calls-for-proposals, etc., for example:

$$\begin{aligned} \text{Perf} &= \{inform, confirm, disconfirm, \ldots, \mathbf{null}, \ldots\} \\ \text{Prot} &= \{negotiate, cfp, \ldots, \mathbf{no\_protocol}, \ldots\} \end{aligned}$$

Note we include a null performative which is a 'do nothing' (no reply) performative (cf. *silence* as used in Smith *et al.* [1998]), and require an empty protocol no...protocol: agents can communicate using one-shot speech acts irrespective of a particular protocol.

Each element of Prot names a finite state diagram. Then *reply* is a (partial) function from performatives, protocols and protocol states to the power set of performatives. This states for each perfomative, 'uttered[1] in the context of a conversation following a specific protocol, what performatives are acceptable replies. The *reply* function can be constructed from inspection of the protocol state diagrams, and vice versa, although more formal characterizations are possible (e.g. [Kuwabara *et al,* 1995]).

This 3-tuple is standard for all agents using the ACL. To fully characterise the semantics, we need three further functions which are relative to an agent a, and specify what an agent does with a message, not how it does it:

$$\begin{aligned} add_a &: \quad \text{Perf} \times \text{Prot} \times content\_meanings \times \Delta \to \Delta \\ select_a &: \quad \wp(\text{Perf}) \times \Delta \to speech\_acts \\ conv_a &: \quad c \mapsto \sigma \end{aligned}$$

Here, add_a is agent a's own procedure for computing the change in its information state* from the content of an incoming message using a particular performative 'uttered[1] in the context of a particular protocol. add_a then converts a's (a-type) information state into a new (a-type) information state, $select_a$ is agent a's own procedure for selecting a performative from a set of performatives (valid replies), and from its current information state generating a complete speech act for this performative which will be its (intended) reply. Finally, we require that agents keep track of the state of each conversation in which it is involved, and uniquely identify each conversation with some identifier, i.e. an agent may be involved in more than one conversation and uses the identifier to distinguish between them. $conv_a$ then maps a conversation identifier onto the current state of the protocol being used to conduct the conversation. (Note that FIPA ACL has message attributes (reply-with, in-reply-to and conversation-id) to serve such purposes, but they are not accommodated in the formal semantics.)

An agent *s* communicates with (and communicates information to) an agent r via a speech act. This is represented by:

$$< s, \mathsf{perf}(r, (C, L, O, p, i, t_{snd})) >$$

This is saying that *s* does (communicates with) performative perf with content *C* in language *L* using ontology 0, in the context of a conversation identified by *i* which is following protocol p, at the time of sending t_{snd}.

Define a function / which for any speech act *sa* returns the performative used in that speech act. The meaning of a speech act is then given by:

$$[\![< s, \mathsf{perf}(r, (C, L, O, p, i, t_{snd})) >]\!] = \mathcal{I}_r < r, sa >$$
$$\text{such that} f(sa) \in reply(\mathsf{perf}, p, conv_r(i))$$

This defines the meaning of the speech act by sender s to be an intention of receiving agent r to perform some other speech act. The performative used in the speech act as the response is selected from $reply(\mathsf{perf}, p, conv_r(i))$, i.e. it is constrained to be one of the performatives allowed by the protocol. The speech act is generated from r's information state at the time of selection $\Delta_{r,t_{now}}$ and the state of the conversation *i* according to the protocol p, using r's $select_r$ function:

$$sa = select_r(reply(\mathsf{perf}, p, conv_r(i)), \Delta_{r,t_{now}})$$

where $\Delta_{r,t_{now}}$ is given by:

$$\Delta_{r,t_{now}} = add_r(\mathsf{perf}, p, L_r[\![C]\!]_{L,O}, \Delta_{r,t_{rcv}})$$

This states that r's information state after receiving the message is the result of adding, using r's own procedure add_r, r's own interpretation (content meaning) of the content *C* in language *L* using ontology *O* to its database at the time of receipt, $\Delta_{r,t_{rcv}}$.

Note that for a speech act performed as part of a conversation identified by i, both sending and receiving agents are expected to update their respective *conv* mapping from conversation identifiers to current state (as defined by the protocol). The sending agent will do this on performing the speech act, the receiving agent after creating the intention to reply. This means that after a message has been sent and before it has been processed, the two parties of the conversation will (for a while) be in different states, and this can be useful in error recovery as a result of lost messages, for example.

It is possible, with a little care, to describe FIPA ACL as a 3-tuple as defined above. FIPA ACL has a small set of basic performatives and just a few protocols, most with fewer than 10 states. How'ever, more work is required for protocols where there are more than two participants, the sequence of speech acts is not simple turn-taking, there are timing constraints which affect allowable replies, and so on. It may also be that a protocol, if not infinite, may nevertheless have a 'very large' number of states. Therefore a further generalisation of the specification may be required, by defining rules for generating speech acts between two players (i.e. a game).

To summarise, all agents should react to a speech act, and we try to constrain and predict the possible reactions with protocols. These, we argue, are the normative standard items: what an agent can do, not how it does it. add_a and $select_a$ are specifying what agent *a* should do, not how it should do it. However, even if an ACL with a standard external semantics can be agreed, it is unlikely to be testable as the agents are complex entities. It does impose a requirement that agents involved in a conversation behave according to a protocol, but 'anti-social'[1] behaviour may not be immediately obvious and the history of communications needs logging, and a means of auditing, policing and accountability is required. Standardising the protocols alone will not achieve this.

4 The Proper Role of Belief States

We have seen how an agent's information state (belief state) can be used to guide the selection of a response to a message, and this formed the core of our proposed semantics for the ACL. This would be a normative specification. How then do agents choose which speech ads to perform in the first place? It is in answer to this question that speech act- theory, as originally conceived, and agent belief states, can contribute to an informative specification. We specify add_a and $select_a$ with respect to beliefs, desires and intentions, i.e. by giving an intentional semantics at level (hi) of Section 2.

For example, reconsider our 'sincerity' axiom:

$$\models \mathcal{B}_s\phi \wedge \mathcal{D}_s\mathcal{B}_r\phi \rightarrow \mathcal{I}_s(\text{DONE}(< s, inform(r, \phi) >))$$

The 'logical operation' of this axiom is closely related to the BDI agent architecture of Kirmy *et al.* [1995], where the combination of beliefs and desires trigger intentions. As an informative specification, it guides agent developers as to the circumstances under which inform speech acts could (or should) be performed, but does not constrain them to use BDI architectures.

We then provide, as an informative specification, that:

$$add_a(\mathsf{inform}, \textbf{no\_protocol}, \phi, \Delta_a) = \Delta_a \cup \{\phi\}$$

which is to say that the basic intuition behind receiving, from a sincere agent, an inform message not in the context of any protocol, is just to add the content to the agent's information state. However, the implementation of add_a for a particular agent *a* need not be so trusting, also it could do 'its own thing' in dealing with additional inferential effects, inconsistency, multiple sources of information, belief revision (non-monotonic logic), copies of *(p* (resource logics), and so on.

We illustrate this idea in the next section. We conclude this section with the observation that the extent to which agents wish to expose their behaviour by publicising their *add* and *select* functions, also defines the extent to which this behaviour can be verified. Otherwise, all that is required, to be compliant to a standard, for example, is that an agent should make an appropriate response to a particular input, as given by the *reply* function, and this can be verified with relative ease.

5 A BDI Implementation

In this section, we discuss an operational model of the BDI agent architecture as suggested in [Kinny *et* al., 1995], enhanced to accommodate BDI-reasoning about agent-agent communication protocols based on the semantics described in the previous section.

5.1 The BDI Architecture

Kinny *et al 's* [1995] BDI-agent architecture consists of the modules illustrated in Figure 2. Here, the belief database contains facts about 'the world'; the desires module contains goals to be realized; the plan library consists of plans, which are sequences of actions which achieve goals; and the intention structures contains instances of those plans chosen for execution (and currently being executed). The interpreter executes intentions, updates beliefs, modifies goals, and chooses plans.

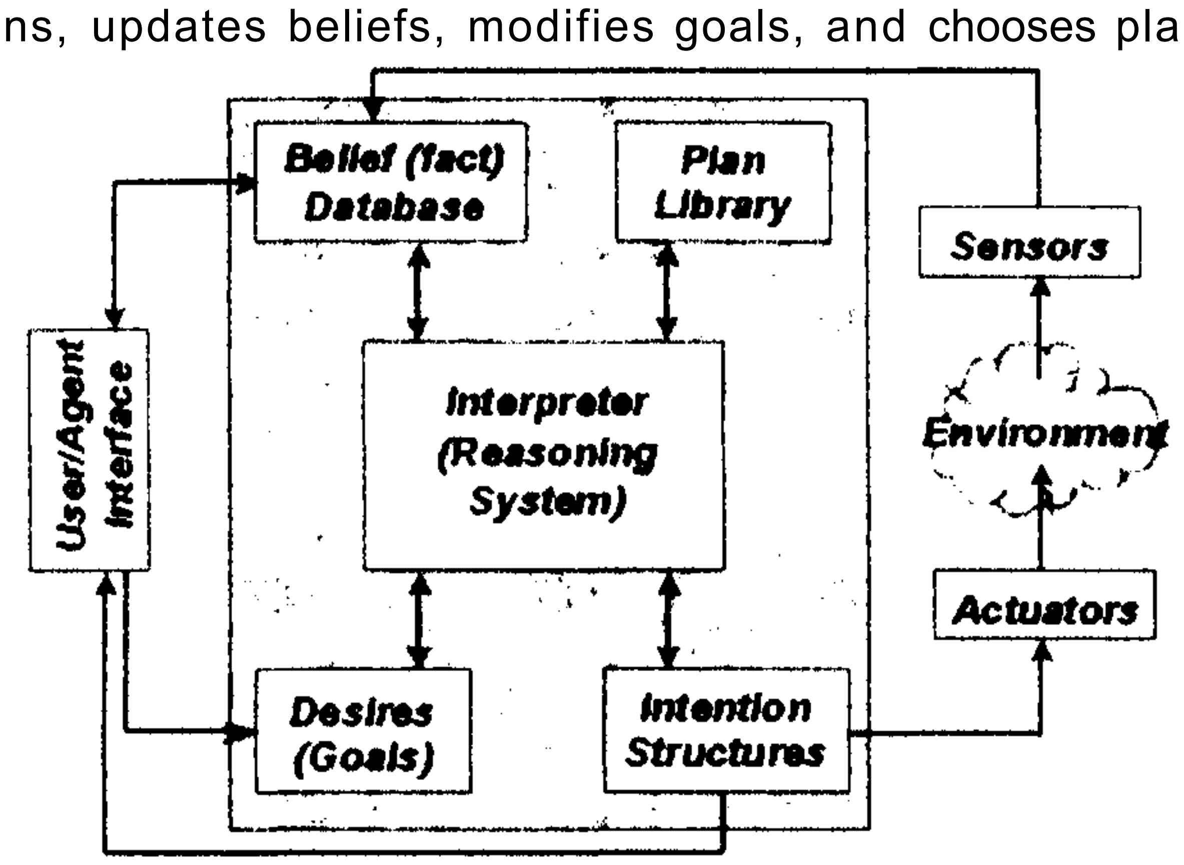

Figure 2: BDI Agent Architecture

The interpreter execution cycle is as follows: At time *t:* certain goals are established, and certain beliefs are held. Event(s) occur that alter the beliefs or modify goals, and the new combination of goals and beliefs trigger action plans. One or more action plans are selected and placed on the intention structures. An executable plan is selected and one step is executed, whereby the agent performs the action. Performing the action changes the environment and may establish new goals and beliefs, and so the interpreter execution cycle starts again.

We now describe how we envisaged the BDI architecture working in conjunction with the ACL semantics, for a pair of communicating agents.

5.2 A Semantic Specification

Consider again the communicating agents in Figure 1. Suppose they are telling each other when they have changed something in the environment in which they are embedded. To ensure that their perceptions of the environment are aligned, they will use a mutually agreed protocol, called *datasync.* This protocol is illustrated as a finite state diagram in Figure 3.

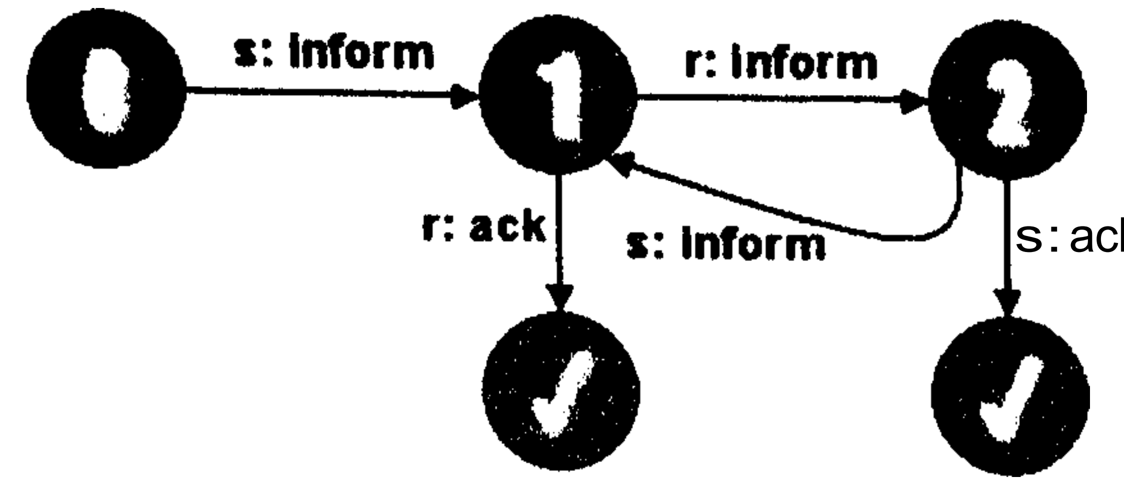

Figure 3: *datasync* Protocol

The idea is for one agent to inform the other agent, of changes in the environment, and for this other agent to agree the change (via the acknowledge speech act ack), or correct it via another inform speech act.

We can formalise this as part of our ACL semantics as follows, by specifying, inter alia:

$$
\begin{array}{rcl}
\mathsf{Perf} & = & \{\mathsf{inform}, \mathsf{ack}, \mathbf{null}, \ldots\} \\
\mathsf{Prot} & = & \{\mathit{datasync}, \mathbf{no\_protocol}, \ldots\} \\
\mathit{reply} & : & \mathsf{inform} \times \mathit{datasync} \times 0 \rightarrow \{\mathsf{inform}, \mathsf{ack}\} \\
 & & \mathsf{inform} \times \mathit{datasync} \times 1 \rightarrow \{\mathsf{inform}, \mathsf{ack}\} \\
 & & \mathsf{inform} \times \mathit{datasync} \times 2 \rightarrow \{\mathsf{inform}, \mathsf{ack}\} \\
 & & \mathsf{ack} \times \mathit{datasync} \times 1 \rightarrow \{\mathbf{null}\} \\
 & & \mathsf{ack} \times \mathit{datasync} \times 2 \rightarrow \{\mathbf{null}\}
\end{array}
$$

The *reply* function therefore specifies the acceptable replies to messages sent in the context of the *datasync* protocol. The protocol is initiated by an inform message, and terminated (with success) by an ack message.

Ignoring for now issues like language, time, ontology and protocol, an agent designer could specify an agent a's behaviour for reacting to an inform message with content ϕ from an agent *s* using the *datasync* protocol to be:

$$
\begin{array}{lll}
\multicolumn{3}{l}{\mathit{add}_a(\mathsf{inform}, \mathit{datasync}, \phi, \Delta_a) = \Delta'_a, \text{ where}} \\
\phi \in \Delta_a & \rightarrow & \Delta'_a = \Delta_a \\
\phi \notin \Delta_a & \rightarrow & \Delta'_a = \Delta_a \cup \{\phi, \mathcal{D}_a \mathcal{B}_s \mathcal{B}_a \phi\} \\
 & & \text{if } \Delta_a \cup \{\phi\} \text{ is consistent} \\
 & \rightarrow & \Delta'_a = \Delta_a \cup \{\phi^c, \mathcal{D}_a \mathcal{B}_s \phi^c\} \\
 & & \text{if } \Delta_a \cup \{\phi\} \text{ is inconsistent} \\
\phi^c \in \Delta_a & \rightarrow & \Delta'_a = \Delta_a - \{\phi^c\} \cup \{\phi, \mathcal{D}_a \mathcal{B}_s \mathcal{B}_a \phi\} \\
 & & \text{if } \mathit{belief\_revise}_a(\Delta_a, \phi, \phi^c) = \phi \\
 & \rightarrow & \Delta'_a = \Delta_a \cup \{\mathcal{D}_a \mathcal{B}_s \phi^c\} \\
 & & \text{otherwise}
\end{array}
$$

This is only an exemplary specification in a semi-formal notation. Note the formula ϕ^c denotes the complement of formula ϕ. It treats each of the three cases: when the content of the inform is already known, new, or contradictory. The function $\mathit{belief\_revise}_a$ is a gloss on a's belief revision for contradictory statements (cf. [Gathers, 1992]), and we are not saying how the agent ensures its database is consistent (but there are algorithms for doing this, e.g. forward chaining).

This intentional reading of informing agent *a* of some information can be paraphrased informally as follows (where all replies are intuitively *intentions,* to be consistent with the ACL semantics):

If you tell me something I already know,
then I'll stay as I am;
If you tell me something I don't know, then
if it is consistent with my database Δ,
I'll add it to Δ *and acknowledge it*
else if it is inconsistent with Δ
I'll inform you that I disagree;
If you tell me something that I disagree with, then
if I prefer your version,
I'll revise Δ *and acknowledge it*
else if I prefer my version
I'll keep Δ *as it is and inform you otherwise*

This is of course just one formulation: the treatment of new and contradictory information may not be treated in the same way, for example. It is also easy to refine such a specification to incorporate elements of trust. Furthermore, since the sincerity condition is predicated on the notion of co-operation; and all speech acts involve the receiver recognizing an intention on the part of the sender, agents are free (but not forced) to make further inferences about the other agent's beliefs and intentions. Different inferences may be more appropriate for particular types of agents in different kinds of application.

However, we can now appreciate the potential utility of intentionality for individual agents: indeed, we can even imagine "publishing' this behavioural interface in an open system. It then becomes effectively a social commitment on the receiving agent's part (ef. [Singh, 1998]). The intriguing possibility then is that the 'standard' behaviour (sought by the FIPA97 specification) could actually be an emergent property of the system. For now, we show how this specification operates in the communicating agents example of section 2, where each agent is (at least specified as) a BDI agent.

5.3 The Semantics in Operation

In the plan library, we posit plan axiom schema of two types, which we call proactive and reactive. The two types of plan both generate intentions. The first type is the type of plan schema that triggers a dialogue, with the agent as the initiator of that dialogue. The second type is the plan schema that an agent uses to reply to messages that have been received, determining, for example, if there were a protocol, which possible responses would conform to the protocol.

We suppose that both agents have plan axiom schema in the plan library, which for agent *a* (where *a* could be sender *s* or receiver r, although it need not be the same for both) are:

$$\mathcal{B}_a\phi \wedge \mathcal{D}_a\mathcal{B}_R\phi \rightarrow \mathcal{I}_a < a, \mathsf{inform}(R, \phi, datasync) >$$
$$[R, \mathsf{inform}(a, \phi, datasync)]\mathcal{B}_a\phi \wedge \mathcal{D}_a\mathcal{B}_R\mathcal{B}_a\phi \rightarrow \mathcal{I}_a < a, \mathsf{ack}(R, \mathcal{B}_a\phi, datasync) >$$
$$[R, \mathsf{inform}(a, \phi, datasync)]\mathcal{B}_a\phi^c \wedge \mathcal{D}_a\mathcal{B}_R\phi^c \rightarrow \mathcal{I}_a < a, \mathsf{inform}(R, \phi^c, datasync) >$$

The first plan axiom schema is proactive and states that if agent *a* believes ϕ and has the goal that another agent *R* share that belief (the preconditions for action), then agent *a* will form the intention to inform *R* of ϕ, using the *datasync* protocol. The other two plan axiom schema are reactive and determine replies. The former states that after some agent i? has informed agent *a* of some information ϕ, if agent *a* believes the proposition and wants the original sender to believe this, then it will form an intention to reply with an acknowledgement. Similarly, if the informed agent disagrees with the content it will inform the originator of this. It should be clear that the latter two axioms specify the decision making for response making in the *datasync* protocol if add_a has been implemented as described above.

We now logically animate this specification for a sending and a receiving agent. For the sending agent .s, in its belief database there are facts, such as *p* (implicitly $\mathcal{B}_s p$), and modal action schema. These state the beliefs that an agent will have after performing an action. For example, for agent ,s, we could have:

$$\begin{array}{lll} \mathcal{B}_s: & p & \text{(fact } p\text{)} \\ & [a, \mathsf{inform}(R, \phi, \_)]\mathcal{B}_R\phi & \text{(after } s \text{ informs } R \text{ of } \phi\text{,} \\ & & s \text{ believes } R \text{ believes } \phi\text{)} \end{array}$$

Here *R* and ϕ are any agent and formula respectively. The post-condition of the modal action schema specifies the intended rational effect of the action which the agent will believe after doing the action (if the agent later discovers it is not true[1], it will need to do belief revision). This formula concerns the inform performative, irrespective of the protocol it is used in.

In the desire database, there are goals. For example, if the goal of *s* is for r to believe p, we have $\mathcal{D}_s : \mathcal{B}_r p$.

Beliefs and desires are used to trigger a plan action schema and create instances which will be executable plans. In the case of *s,* the following simple plan will bo placed on the intention structures:

$$\mathcal{I}_s :< s, \mathsf{inform}(r, p, datasync) >$$

When the intention is chosen for execution by the interpreter, the intention is fulfilled, the belief state changes according to the appropriate instance of the action axiom schema, i.e. $[s, \mathsf{inform}(r,p)]\mathcal{B}_r p$, the goal is discharged and withdrawn from the desire database, so we end up, in this case, with:

$$\begin{array}{lll} \mathcal{B}_s: & p & \mathcal{D}_s : \emptyset \quad \mathcal{I}_s : \emptyset \\ & \mathcal{B}_r p & \\ & [a, \mathsf{inform}(R, \phi,)]\mathcal{B}_R\phi & \end{array}$$

This is the situation with agent *s.* For agent r, who receives the message, the database prior to receipt might have been entirely empty:

$$\mathcal{B}_r : \emptyset \quad \mathcal{D}_r : \emptyset \quad \mathcal{I}_r : \emptyset$$

Now it receives the message $< s,$ inform(r, *p, datasync)* $>$. The interpreter of r runs its procedure add_r on this input and this database; being a new conversation, the protocol state is 0. If add_r is implemented as specified above, then r will add p to its belief database (p is not present and adding it is consistent), and generate the goal for the sender to believe that it believes it:

$$\mathcal{B}_r : p \quad \mathcal{D}_r : \mathcal{B}_s\mathcal{B}_r p \quad \mathcal{I}_r : \emptyset$$

From the reactive plan axiom schema in the plan library, after an inform r will have the intention to send an acknowledge or an inform. The BDI interpreter is effectively applying the $select_r$ function, and as with the sending agent beliefs and desires are used to trigger a plan action schema and create instances which will be executable plans. The appropriate intention is placed on the intention structures:

$$\mathcal{B}_r : p \quad \mathcal{D}_r : \mathcal{B}_s\mathcal{B}_r p \quad \mathcal{I}_r :< r, \mathsf{ack}(s, p, datasync) >$$

Now we see that the meaning of the speech act is indeed the intention to reply with a valid performative in the context of a protocol, as specified in sections 3 and 4.

6 Conclusions and Further Work

With growing investment in open, agent-based system design and deployment, there is a need for software standards defining the interface between agents. The F1PA standardisation body is probably the most important recent development in the agents field and offers significant potential advantages for developers of open, heterogeneous, and interoperable agent systems.

However, there are considerable limitations of using mental attitudes for standardising the semantics of the ACL [Singh, 1998]. Intentionality is concerned with agent, internals and a communicative act is an external phenomenon: mental attitudes only give a reason for and not the meaning of the performatives. They can give guidance to developers but may be too strong a constraint for heterogenous agents in varying applications.

The protocols specified by FIPA can give purpose and meaning to individual performatives but only in the context, of a conversation that has an identifiable objective. However, there appears to be a growing consensus that conversations, protocols, and social context are the important factors to consider in defining an ACL's formal semantics [Singh, 1998; Labrou and Finin, 1998; Smith *et* al., 1998].

Our own experience of applying the FIPA specifications suggests that interactions in a multi-agent system could be designed on the basis of the standard performatives and protocols. This work has gone one step further and formalised the normative use of performatives in specific protocols. Furthermore, the relationship between what an agent 'thinks' about the content of a message to what it 'does' in response to receiving that message is formalised through informative specifications expressed in terms of beliefs, desires and intentions.

The semantic framework described here actually defines a *class* of ACLs. The challenges ahead then include: extending an ACL specification with new protocols and performatives, and customising with intentional specifications for particular applications; describing conversation states by structures which are affected by speech acts; allowing for multi-party conversations; identifying a way for agents to publicise thier understanding and use of new performatives and protocols (so-called brown pages); and providing a mechanism for agents to discover and learn a new dialect. The issues of social behaviour and time also need to be addressed as these are both significant factors in why and when agents communicate

In the meantime, we have submitted this work to FIPA as a proposal for the ACL semantics in FIPA99.

Acknowledgements

This research has been undertaken in the context of the UK EPSRC/Nortel Networks joint funded project CASBAh (GR/L34440) and the EU funded ACTS project MARINER (AC333). We are grateful for the constructive comments of the anonymous reviewers.

References

[Breiter and Sadek, 1996] P. Breiter and M. Sadek. A rational agent as a kernel of a co-operative dialogue system. In *Proceedings EC AT 96 ATAL Workshop,* pp261 276. Springer-Verlag. 1996.

[Cohen and Levesque, 1995] P. Cohen and H. Levesque. Communicative actions for artificial agents. In V. Lesser, ed.. *Proceedings ICMAS95.* AAAI Press, 1995.

[Finin *et* al., 1995] T. Finin, Y. Labrou, and J.Mayfield. KQML as an agent communication language. In J. Bradshaw, ed., *Software Agents.* MIT Press, 1995.

FIPA, 1997] FIPA. FIPA'97 Specification Part 2: ACL. FIPA, http://drogo.cselt.stet.it/fipa/, 1997.

[Galliers, 1992] ,]. Galliers. Autonomous belief revision and communication. In P. Gardenfors, ed.. *Belief Revision.* Cambridge University Press, 1992.

Kiniiy *et* a/., 1995] D. Kinny, M. Georgeff, J. Bailey, D. Kemp, and K. Ramamohanarao. Active databases and agent systems: A comparison. In *Proc. 2nd International Rules in Database Systems Workshop.* 1995.

Kuwabara *et* al., 1995] K. Kuwabara, T. Ishida, and N. Osato. AgenTalk: Describing multiagent coordination protocols with inheritance. In *Proc. 7th IEEE 1CTAI95,* pp46() 465. 1995.

[Labrou and Finin, 1998] V. Labrou and T. Finin. Semantics and conversations for an agent communication language. In M. Huhns and M. Singh, eds., *Readings in Agents,* pp235 242. Morgan Kaufmann, 1998.

Singh, 1998] M. Singh. Agent communication languages: Rethinking the principles. *IEEE Computer,* pp40 47, December, 1998.

Smith *et* al., 1998] I. Smith, P. Cohen, J. Bradshaw, M. Greaves, and H. Holmback. Designing conversation policies using joint intention theory. In Y. Demazeau, ed., *Proceedings ICSMAS98.* 1998.

[Wooldridge, 1998] M. Wooldridge. Verifiable semantics for agent communication languages. In Y. Demazeau, ed., *Proceedings ICMAS98.* 1998.