Communicative Actions for Artificial Agents

Philip R. Cohen
Center for Human-Computer Communication
Dept. of Computer Science and Engineering
Oregon Graduate Institute of Science and Technology
and
Hector J. Levesque
Dept. of Computer Science
University of Toronto

June 19, 1995

1 Introduction

A language for interagent communication should allow agents to enlist the support of others to achieve goals, to commit to the performance of actions for another agent, to monitor their execution, to report progress, success, and failure, to refuse task allocations, to acknowledge receipt of messages, etc. Crucially, a collection of agents needed to accomplish a task will frequently include humans who have delegated tasks to the agents, and/or humans who will be performing some of the work. As such, it is essential that the functions being offered by the communication language be common across the language of intelligent agents and the language that people will use to communicate with them.

It so happens that there is such a language, the language of "speech acts" [3, 25], or more precisely, "illocutionary acts." Such actions include requesting, promising, offering, acknowledging, proposing, accepting, etc. Philosophers of language noted that human utterances are the observable byproduct of such actions, and moreover, that utterances may realize more than one such action simultaneously (such as being both an assertion and a request). AI researchers have modeled such actions as operators in planning systems [1, 2, 6, 14] and have developed logical frameworks for providing their semantics [12, 22, 24].

Recently, a number of researchers have proposed artificial languages based on speech act theory as the foundation for interagent communication [15, 18, 19, 27, 28]. The most elaborate and developed of these is KQML [15]. In this language, agents communicate by passing so-called "performatives" to each other. KQML is offered to the agent community as an *extensible* language with an open-ended set of performatives, whose meaning is independent of the propositional content language (e.g., Prolog, first-order logic, SQL, etc.) However, the authors of KQML have yet to provide a precise semantics for this language, as is customary with programming languages.[1] Without one, agent designers cannot be certain that the interpretation they are giving to a "performative" is in fact the same as the one some other designer intended it to have. Moreover, the lack of a semantics for communication acts leads to a number of confusions in the set of reserved "performatives" supplied. Lastly, designers are left unconstrained and unguided in any attempt to extend the set of communication actions.

This paper claims that substantive confusions exist in the KQML specification that undermine its semantical basis. As an example of how semantics can be given to communicative actions, we propose adequacy criteria for a semantical treatment, and illustrate how our semantics of speech acts obeys them. Finally, we discuss the impact these analyses may have on various design decisions made in KQML.

2 Background: Performatives

In natural languages, a performative utterance is one that succeeds simply because the speaker *says* or *asserts* s/he is doing so. Usually, in English, such utterances arise in a first-person, present tense declarative utterance, often accompanied by "hereby", as in "I hereby request you to get off my foot." However, this is not always the case, and performatives can occur in the third person ("We request you to attend the marriage of our son", in the passive ("Passengers are requested to refrain from smoking"), and even brokered via third parties (as when a translator might say "The King requests you to remove your shoes.") The important commonality among these uses is that the action is performed by saying so. Essentially, the speaker is *asserting* that *this* utterance is a request, order, or whatever.

When does saying so make it so? There are a number of cases to be considered, but we only discuss one here (for a more complete discussion, see [11]). Performative uses of illocutionary verbs succeed because, on our analysis, such verbs are defined as *attempts*. That is, the semantics of the illocutionary act is that the speaker is attempting to communicate his/her mental state. By asserting that s/he is doing so, the act succeeds in doing so because the speaker is taken to be an expert on his or her own mental state. In most cases, then, the listener will immediately assume the speaker is in the requisite mental state, and the act succeeds. (See [11] for proofs of the relevant theorems.) Importantly, it should be noticed that many verbs, namely, perlocutionary verbs, *cannot* be used performatively. For example, in English, "I hereby convince you that your birthday is March 3" cannot be a performative utterance because one cannot be guaranteed that the listener will be convinced.

A second constraint to be observed is that some verbs are *self-defeating* when used performatively. For example, "I hereby lie to you that I took out the garbage" cannot be a successful lie. Our semantics shows logically why this cannot be so, and thus provides a model for the kinds of constraints that can apply on the formation of new performatives.

[1] A first attempt has been made [18], but much work remains.

The problem of performatives for natural language speech act theories, then, is how an utterance in declarative mood, which looks like an assertion, can in fact (also) be something else, namely the act it names. That is, what is uttered appears to be something with a truth value. But most illocutionary acts, for example, requests, do not have a truth value. Rather, what has a truth value is the proposition that *this* utterance constitutes the named action.

3 KQML

Briefly, KQML is a communication language that has been designed to facilitate high-level cooperation and interoperation among artificial agents [15, 16, 17, 18]. Agents may range from simple programs and databases to more sophisticated knowledge-based systems. The language is proposed as a general-purpose standard for interagent communication [18], and is being used in a number of projects in the U.S.A. KQML offers an extensible set of so-called "performatives" that specify what kinds of communications agents can have with one another. Examples include ACHIEVE, ADVERTISE, ASK-IF, ASK-ALL, BROKER, DENY, ERROR, STREAM-ALL, TELL, UNACHIEVE. For example, the following is the description of the "performative" DENY, taken from [15]:

```
deny
    :content  <performative>
    :language KQML
    :ontology <word>
    :in-reply-to <expression>
    :sender <word>
    :receiver <word>
```

> Performatives of this type indicate that the meaning of the embedded `<performative>` is *not* true of the sender. A `deny` of a`deny` cancels out.

Communicative actions such as these are considered performatives in the sense that "the message is intended to perform some action in virtue of being sent" [15, p. 4].

Coupled with KQML is a set of policies that dictate constraints on legal sequences of communication acts, which are enforced through some combination of constraints on KQML developers, agent developers, and specific modules (e.g., the conversation module [18]). These policies induce a set of interagent conversation patterns using the communication actions.

4 Critique of KQML as an agent communication language

We have identified three general difficulties with the draft KQML specification [15].

Ambiguity and vagueness. The meaning of the reserved or standard performatives is rather unclear. Performatives are given English glosses, which often are vague or ambiguous. For example, given that the definition of DENY says that the embedded performative is not true of the speaker, if an agent DENYS a TELL, does that mean the agent did not TELL earlier, or does not believe what is being said now? A close reading of this definition reveals another confusion lurking — it says that what agents deny is a performative, and it is no longer *true* of the speaker. This implies that performatives *do* in fact have truth values, and are not actions after all. If so, then the semantic type of a performative is most likely a proposition. We therefore return to the problem of natural language performative utterances, namely how the uttering of an expression with a truth value constitutes the performance of an action. Actually, we do not believe this interpretation is intended by the authors of KQML. Rather, the definition of DENY, as well as those of many other "performatives," are simply miscast. However, given that other performatives are defined in terms of DENY, such as UNREGISTER and UNTELL, it is no small error.[2]

Misidentified performatives. KQML labels as performatives actions that should be the argument of a directive speech act, e.g., requesting. For example, ACHIEVE is glossed in English as a request to get the addressed agent to make something true. But, the formalism omits the request. Accordingly, it is an error to include acts such as ACHIEVE, BROKER, STREAM-ALL as performatives because an agent cannot execute another agents' actions or satisfy another agent's goals, merely by saying so (i.e., sending a message). In fact, the relevant performative should be a directive act (e.g., a request).

In summary, there are really only two types of speech acts discussed in the draft specification for KQML, the directive and assertive. Semantics of these actions has been given in great detail elsewhere [12, 24], and will be briefly discussed below.

Missing performatives. Although KQML offers an extensible language, a most important class of communication actions seems to be missing entirely — the commissives, which commit an agent to a course of action. The prototypical example of a commissive is promising; other examples include accepting a proposal, and agreeing to perform a requested action. Without these actions, it is hard to see how any multiagent system could work robustly. Whenever an agent is busy and cannot immediately execute a requested action, the requesting agent would be blocked, as it has no reason to believe that it can proceed, and that an answer will be forthcoming. The requesting agent cannot even receive a confirmation that its requested action has been accepted by a given agent, as there are no acceptances in the language.[3]

[2] Labrou and Finin [18] propose a different definition of `Deny`, in which the senders essentially states that it does not believe a proposition. According to this semantics, the language of KQML has changed dramatically, to include embedded propositions instead of embedded performatives.

[3] We are aware that numerous researchers are attempting to add new performatives, many in the commissive class. Our point is that the basic language should provide the generic commissive speech act, which

Some researchers have suggested that the language does not need commissive performatives. Rather, it is argued that a TELL that the sender *will* do an action A should suffice. We believe there are numerous problems with this suggestion. First, the logical form of this TELL is essentially that of a prediction — the sender is asserting what *will* be true in the future. When a prediction fails, it is open to the agent merely to say "well, I was wrong," and to revise its beliefs. This is simply too weak. Someone who proposes marriage and hears the performative "I assert that I will marry you (eventually)" in response is unlikely to rent a wedding hall.[4] A commitment, however, would be cause for celebration — the agent is supposed to *make* it true, to screen out incompatible options, to track its success, etc. [5, 10].

The second option is to approximate a commitment with an agent's telling what *it* will do. The only way this can succeed is if we insist that agents do not come to this conclusion in any way other than by having a commitment/intention. Otherwise, the agent may fall victim to the "Little Nell" problem [10, 20] in which the agent never forms the intention to act because it already believes the action will happen. But, if the conclusion that the agent eventually acts arises because the agent is committed, why not have the agent just say so?

The third difficulty with trying to avoid commitment by issuing a TELL of what the sender will do, is that acknowledging this special case would violate the KQML design decision of having the performatives be independent of the content language. Here, one would have to stipulate special rules for contents of the apppropriate form (i.e., a future operator applied to the agent's actions).

It appears to us that KQML is underspecified, and in need of a semantical treatment to clarify what in fact agents and agent designers are supposed to mean by performing communicative actions. This lack of a semantics has not gone unnoticed, as Labrou and Finin [18] have attempted an initial semantics using a simple pre/post-condition analysis of speech acts, loosely based on Searle and Vanderveken's multidimensional semantics [26]. We have discussed elsewhere [12] problems with this type of semantics, specifically that much of what is stipulated can be derived. As an example, of the *kind* of semantics we believe is called for, we review our analysis of speech acts, which is appropriate to both human and artificial agents. Although we provide semantics in terms of mental states, the agents themselves need not reason using these attitudes, but need only behave according to the principles they entail.

We begin by giving an abbreviated description of our analysis of rational action upon which we erect a theory of speech acts. The theory is cast in a modal logic of belief, goal, action, and time. Further details of this logic can be found in [10]. Then, we show how the speech acts of requesting and informing compose, using as an example, the asking of a yes-no question.

---

other agent designers should specialize for their needs.

[4]The addition of 'eventually' here is logically unnecessary, but serves to draw the distinction more clearly.

5 Abbreviated theory of rational action

5.1 Syntax

The language we use has the usual connectives of a first-order language with equality, as well as operators for the propositional attitudes and for talking about sequences of events: (BEL x p) and (GOAL x p) say that p follows from x's beliefs or goals (a.k.a choices) respectively; (BMB x y p) says that x believes that p is a mutual belief with y; (AGT x e) says that x is the only agent for the sequence of events e; $e_1 \leq e_2$ says that e_1 is an initial subsequence of e_2; and finally, (HAPPENS a) and (DONE a) say that a sequence of events describable by an action expression a will happen next or has just happened, respectively.

An action expression here is built from variables ranging over sequences of events using the constructs of dynamic logic: a;b is action composition; a|b is nondeterministic choice; a||b is concurrent occurrence of a and b; p? is a test action; and finally, a^* is repetition. The usual programming constructs such as IF/THEN actions and WHILE loops, can easily be formed from these. Because test actions occur frequently in our analysis, yet create considerable confusion, read p?;a as "action a occurring when p holds," and for a;p?, read "action a occurs after which p holds." We use e as a variable ranging over sequences of events, and a and b for action expressions.

We adopt the following abbreviations:

$$\text{(DONE x a)} \stackrel{\text{def}}{=} \text{(DONE a)} \wedge \text{(AGT x a)}.$$

$$\text{(HAPPENS x a)} \stackrel{\text{def}}{=} \text{(HAPPENS a)} \wedge \text{(AGT x a)}.$$

$$\text{(AFTER a p)} \stackrel{\text{def}}{=} \text{(HAPPENS a;p?)}$$

$$\Diamond \text{p} \stackrel{\text{def}}{=} \exists \text{e (HAPPENS e;p?)}.$$

$$\text{(LATER p)} \stackrel{\text{def}}{=} \neg \text{p} \wedge \Diamond \text{p}.$$

$$\Box \text{p} \stackrel{\text{def}}{=} \neg \Diamond \neg \text{p}.$$

$$\text{(PRIOR p q)} \stackrel{\text{def}}{=} \forall \text{c (HAPPENS c;q?)} \supset \exists \text{a (a} \leq \text{c)} \wedge \text{(HAPPENS a;p?)}.$$

The proposition p will become true no later than q.

$$\text{(KNOW x p)} \stackrel{\text{def}}{=} \text{p} \wedge \text{(BEL x p)}.$$

6 Individual Commitments and Intentions

To capture one grade of commitment that an agent might have toward his goals, we define a persistent goal, P-GOAL, to be one that the agent will not give up until he thinks certain conditions are satisfied. Specifically, we have

Definition 1 *Internal Commitment:*
$(\mathsf{P\text{-}GOAL\ x\ p\ q}) \stackrel{\text{def}}{=}$
$(1)\ (\mathsf{BEL\ x\ \neg p})\ \wedge$
$(2)\ (\mathsf{GOAL\ x\ (LATER\ p)})\ \wedge$
$(3)\ [\mathsf{KNOW\ x\ (PRIOR\ [(BEL\ x\ p)} \vee \mathsf{(BEL\ x\ \Box\neg p)} \vee \mathsf{(BEL\ x\ \neg q)]}$
$\qquad \neg[\mathsf{GOAL\ x\ (LATER\ p)}])].$

That is, the agent x believes p is currently false, chooses that it be true later, and knows that before abandoning that choice, he must either believe it is true, believe it never will be true, or believe q, an escape clause (used to model subgoals, reasons, etc.) is false.

Intention is a kind of persistent goal in which an agent commits to having done an action, in a particular mental state.[5]

Definition 2 *Intention:*
$(\mathsf{INTEND}_1\ \mathsf{x\ a\ q}) \stackrel{\text{def}}{=}$
$\quad (\mathsf{P\text{-}GOAL\ x\ [DONE\ x\ (BEL\ x\ (HAPPENS\ a))?;a]\ q}).$

Intending to do an action a or achieve a proposition p is a special kind of commitment (i.e., persistent goal) to having done the action a or having achieved p. However, it is not a simple commitment to having done a or e;p? for that would allow the agent to be committed to doing something accidentally or unknowingly. Instead, we require that the agent be committed to arriving at a state in which he believes he is about to do the intended action next.

This completes a brief discussion of the foundational theory of intention and commitment. Next, we proceed to define the communicative actions.

7 Illocutionary Acts as Attempts

Searle [25] points out that an essential condition for a request is that the speaker be attempting to get the addressee to perform the requested action. We take this observation one step further and define all illocutionary acts as attempts, hence defined in terms of the speaker's mental states. Attempts involve both types of goal states, GOAL (merely chosen) and INTEND (chosen with commitment):

Definition 3 $\{\mathsf{ATTEMPT\ x\ e}\ \Psi\ \Phi\} \stackrel{def}{=}$
$\quad [(\mathsf{GOAL\ x\ (LATER}\ \Psi))\ \wedge$
$\qquad (\mathsf{INTEND}_1\ \mathsf{x\ e;}\Phi?\ (\mathsf{GOAL\ x\ (LATER}\ \Psi)))]?\mathsf{;e}$

That is, an attempt to achieve Ψ via Φ is a complex action expression in which x is the agent of event e and just prior to e, the agent chooses that Ψ should eventually become true, and intends that e should produce Φ relative to that choice. So, Ψ represents some ultimate goal that may or may not be achieved by the attempt, while Φ represents what it takes to make an honest effort.[6]

[5] In [10], we define two types of intention, those to do actions and those to achieve propositions

[6] In a more adequate analysis, there should be a stronger relationship, say contingent causality, between Φ and Ψ.

7.1 Adequacy Criteria for a Semantics for Communicative Actions

In Cohen and Perrault [14] we provided a strong adequacy criterion for theories of human speech acts, namely compositionality. Speech acts that take actions as arguments, such as requests, should be able to take other speech acts as arguments. Moreover the appropriateness and success conditions of the composite communicative act should be based on those of the elementary ones. Thus, yes-no and wh-questions should be correctly handled as composite speech acts, with the proper semantics derived from both request and inform [14, 23, 24]. Similarly, a request should properly embed a request to an intermediating agent [14], and provide reasonable conditions on the knowledge of the intermediating agent about the ultimate recipient. Appropriateness conditions that arise include whether or not the intermediating agent has to believe the recipient can in fact do the requested action. Intuitively, if the original request is simply forwarded to a named recipient, the answer should be "No." However, if the middle agent (i.e., facilitator) is supposed to *recruit* an agent to satisfy a request directed at no one in particular, then the answer should be "Yes."

Below we provide a semantics forillocutionary acts that provides such compositionality. For space considerations, we consider only the case of a yes/no question. We first define speech acts that are appropriate to both human and software agents, and then we discuss where the two kinds of communicative actions may diverge.

7.2 Definitions of Request and Inform

Illocutionary acts (IAs) are modeled as complex action expressions in the dynamic logic. Essentially, an IA is an event performed in the "right" circumstances, namely when the agent is *attempting* to achieve a certain type of effect — to communicate its mental state.

To characterize a request or, for that matter, any illocutionary action, we must decide on the appropriate formulas to substitute for Φ and Ψ in the definition of an attempt. We constrain illocutionary acts to be those in which the speaker is committed to *understanding*, that is, to achieving a state of BMB that he is in a certain mental state. Moreover, we will for now adopt the strong assumption that agents that are part of the network are in fact sincere, although the semantics provided does not depend on sincerity. Our definition of sincerity from [12] is:

Definition 4 *Sincerity* —
(SINCERE x y p) $\stackrel{def}{=}$
  ∀e (GOAL x (HAPPENS x e; (BEL y p)?)) ⊃
    (GOAL x (HAPPENS x (KNOW y p))?)

In other words, agent x is sincere to y about p if whenever x wants y to come to believe p, x wants y to come to know p.

Below is a definition of a speaker's requesting an addressee to achieve α, where α is an action expression containing a free variable e'. The idea is that requesting α is requesting the action to be performed for some value of the free variable. For example, a request to perform the action {ClosedWindow?;e';OpenWindow?} is a request to open the window. Complete definitions and an extensive justification of the various conditions can be found in [12].

Definition 5 {REQUEST spkr addr e α} $\stackrel{\text{def}}{=}$

```
{ATTEMPT spkr e ∃e' (DONE addr α)
    [BMB addr spkr
        (GOAL spkr ∃e'
            [◇(DONE addr α) ∧
              (INTEND₁ addr α
                (GOAL spkr
                  [◇(DONE addr α) ∧
                  (HELPFUL addr spkr)]))])]}
```

That is, event e is a request if it is an attempt at that time to get the addressee to do α, while being committed to making public that the speaker wants (a) that α be done, and (b), that the addressed party should intend to achieve it relative to the speaker's wanting it and relative to the addressee's being helpfully disposed towards the speaker.[7] This means that the addressee is allowed to "get off the hook" if the speaker changes its mind about desiring the action α done.

The illocutionary act of informing can be defined as an attempt get the addressee to know that some proposition is true:

Definition 6 {INFORM spkr addr e p} $\stackrel{\text{def}}{=}$

```
{ATTEMPT spkr addr e
  (KNOW addr p)
   [BMB addr spkr
      (P-GOAL spkr
          (KNOW addr (KNOW spkr p)))]}
```

So an inform is defined as an attempt in which to make an "honest effort," the speaker is committed to making public that he is committed to the addressee's knowing that he knows p. That is, just like a request, the speaker is committed to the addressee's knowing what mental state he is in. Although he is committed to getting the addressee to believe something about his *goals*, what he hopes to achieve is for the addressee to come to know p.

At this point in our analysis of human speech acts, utterance events were characterized as producing a BMB that the speaker is in the mental state characterized by the mood indicators in the utterance (e.g, imperative or declarative). However, the general framework for artificial agents differs somewhat from our analysis of the human case Our notion of an

[7] A *command* speech act would be identical to a request except that instead of involving helpfulness, the intent would be relative to some sort of authority relationship between the speaker and the addressee. 'Helpfulness' is defined in [10].

attempt included a commitment to making an "honest effort," which requires an agent to overcome obstacles to successful delivery and understanding of the message. Let us assume that the KQML framework, with its routers and matching of content languages between the sending and receiving agents, functions as specified. We therefore make the further assumption that the second condition in the attempt, the one the agent is committed to, in fact becomes true after the act.[8]

7.3 Semantics for a Yes-No Question

Using these speech acts, it is now easy to characterize a yes-no question. We simply use the following action expression (suggested in [24]):

$\{\mathsf{REQUEST\ spkr\ addr\ e}$

$\{\mathsf{INFORM\ addr\ spkr\ e'\ p}\} \mid \{\mathsf{INFORM\ addr\ spkr\ e'}\ \neg \mathsf{p}\}\}$

That is, using the nondeterministic choice operator (a disjunction for actions), we have a request for the addressee to either inform the speaker that p or inform the speaker that ¬p.[9] Notice that we have not needed to postulate a new species of inform, namely INFORMIF as done in [1, 14], or ASK-IF in KQML.

Now, among the various felicity conditions for use of this composite action, we need to identify the sincerity condition. We show that in x's asking the question to y whether or not p, the asking agent, is in the following mental state:

$$(\mathsf{GOAL\ x}\ \Diamond(\mathsf{Know\ x\ p}) \vee (\mathsf{Know\ x}\ \neg\mathsf{p}))$$

Theorem 1 *Questions:*

$$\begin{array}{l}(\mathsf{Done\ x}\ \{\mathsf{REQUEST\ x\ y\ e}\\ \qquad \{\mathsf{INFORM\ y\ x\ e'\ p}\} \mid \{\mathsf{INFORM\ y\ x\ e'}\ \neg\mathsf{p}\}\ \}) \supset\\ \qquad (\mathsf{GOAL\ x}\ \Diamond[(\mathsf{KNOW\ x\ p}) \vee (\mathsf{KNOW\ x}\ \neg\mathsf{p})])\ .\end{array}$$

Proof Sketch:

Given what was discussed above, we assume that the network has in fact functioned as designed, and that the requesting event has been observed, understood, and its reception acknowledged by the receiver. Thus, we assume that

$$\begin{array}{l}(\mathsf{BMB\ y\ x}\\ \quad (\mathsf{GOAL\ x}\\ \quad\quad \exists \mathsf{e'}[\Diamond(\mathsf{DONE\ y}\ \alpha)\ \wedge\\ \quad\quad\quad (\mathsf{INTEND}_1\ \mathsf{y}\ \alpha\\ \quad\quad\quad\quad (\mathsf{GOAL\ x}\ [\Diamond(\mathsf{DONE\ y}\ \alpha)\ \wedge\\ \quad\quad\quad\quad\quad\quad (\mathsf{HELPFUL\ y\ x})]))]))\end{array}$$

where α is $\{\mathsf{INFORM\ y\ x\ e'\ p}\} \mid \{\mathsf{INFORM\ y\ x\ e'}\ \neg\mathsf{p}\}$

[8] Note that we could derive conditions under which it comes true, as in [10].

[9] The case of "I don't know" would handled by the need to discharge the standing joint intention [13] that underlies the agent architecture, namely that agents jointly intend to interact.

That is, the receiver y thinks it is mutually believed that the sender x wants y to inform x that p or inform x that $\neg\mathsf{p}$, and to intend to do so relative to x's desires and y's helpfulness.

Let us consider only the first conjunct of the BMB, namely that the speaker wants α to be done. What follows applies to each case separately, embedded within the $(\mathsf{GOAL\ x} \diamond \ldots)$. Specifically, we consider next the performance of α, after which the following holds:

$$\begin{array}{l}[\mathsf{BMB\ x\ y\ (P\text{-}GOAL\ y\ [KNOW\ x\ (KNOW\ y\ p)])}] \vee \\ (\mathsf{BMB\ x\ y\ (P\text{-}GOAL\ y\ [KNOW\ x\ (KNOW\ y\ \neg p)])})\end{array}$$

Because knowledge entails truth,

$$\models [\mathsf{KNOW\ x\ (KNOW\ y\ p)}] \supset [\mathsf{KNOW\ x\ p}]$$

therefore, we have

$[\mathsf{BMB\ x\ y\ (P\text{-}GOAL\ y\ [KNOW\ x\ p]} \wedge \mathsf{[KNOW\ y\ p])}]$, and analogously for informing that p is false.

Now, not only was the "honest effort" condition true, but by the definition of attempting, it was also intended. Therefore, the above condition was intended after $\mathsf{e'}$. In other words, we have, roughly,

$$\begin{array}{l}(\mathsf{INTEND_1\ y\ e'};\\ \quad (\ [\mathsf{BMB\ x\ y\ (P\text{-}GOAL\ y\ (KNOW\ x\ p))}] \vee \\ \quad [\mathsf{BMB\ x\ y\ (P\text{-}GOAL\ y\ (KNOW\ x\ \neg p))}]\ )?\ \ldots)\end{array}$$

Assuming agent y is sincere, we conclude that

$$\begin{array}{l}(\mathsf{INTEND_1\ y\ e'};\\ \quad (\ [\mathsf{KNOW\ x\ (P\text{-}GOAL\ y\ (KNOW\ x\ p))}] \vee \\ \quad [\mathsf{KNOW\ x\ (P\text{-}GOAL\ y\ (KNOW\ x\ \neg p))}]\ ))\ ,\end{array}$$

because a sincere y would not intend for x to believe something falsely. Furthermore, our analysis of the relationship between $(\mathsf{P\text{-}GOAL\ y\ q})$ and $\diamond\mathsf{q}$ (described in Theorem 4.5 in [10, p. 239] indicates that this conclusion can be drawn provided that y remains competent about q (i.e., whenever y comes to believe q, it is correct), and y does not drop its goal to achieve q believing it never will be true. Thus, if we assume y does not come to the conclusion that $(\mathsf{KNOW\ x\ p})$ lightly, and does not come to the conclusion that x will never know that p, x will eventually come to know that p. Hence, we derive that
$(\mathsf{KNOW\ x} \diamond[\ (\mathsf{KNOW\ x\ p}) \vee (\mathsf{KNOW\ x\ \neg p})])$,
which entails $\diamond[(\mathsf{KNOW\ x\ p}) \vee (\mathsf{KNOW\ x\ \neg p})]$.[10]

Given that these conclusions were drawn within the operator $(\mathsf{GOAL\ x} \diamond \ldots)$, we reach the conclusion that the requestor in fact wants to know whether or not p.$\Box$

[10] Now that we have moved back to the level that analyzes what x has chosen, we note that the assumptions that gate the conclusion of $\diamond\mathsf{p}$ from the persistent goal are actually chosen conditions. That is, the x is choosing that y be competent about x's knowing p, and chooses that y does not drop its goals too soon.

7.4 Summary

We have shown how one can determine the sincerity condition of a composite communicative action by combining the definitions of the composed acts.[11] Similar kinds of analyses can be performed for other composed performatives, of which KQML offers many. For example, the so-called ACHIEVE performative should be analyzed as a REQUEST to achieve p, or in our framework, a request to do an e such that $\neg$p?;e;p?. To model the contract net protocol [30], a BID action first needs to be defined, after which one can analyze REQUEST for BID.

As a last example, a proper semantics should distinguish among various kinds of requests, such as requests to REQUEST, and BROKER, RECRUIT, and FORWARD (in KQML). A third-party request involves one agent's requesting another to then request a third agent to perform an action [14]. Because of the semantics we have given, the middle agent will need to have various beliefs and goals about the ultimate recipient. In KQML, a REQUEST to FORWARD differs from a REQUEST to RECRUIT in that the first requires the requestor to know what agent should receive the forwarded action. The second requires the recipient to find an agent to perform the requested action, at the same time requesting that agent to perform the action. Thus, FORWARD would seem to be more specific than RECRUIT. But, in the KQML draft specification, RECRUIT is described as an instance of FORWARD, differing only in the originating agent's specifying the ultimate agent. It would seem, then, that a semantics should provide a common definition, differing only in the scope of a quantifier into a modal operator (if that is how one represents "knowing who".) However, the REQUEST to RECRUIT also shares with the REQUEST to REQUEST the effect that the ultimate recipient is asked to perform some action. Thus, perhaps RECRUIT should be a species of REQUEST. Finally, to complicate matters further, FORWARD seems not to be a species of REQUEST in that an agent can FORWARD a message without the agent's having any beliefs or goals regarding the receiver's subsequent actions. Thus, there are arguments that RECRUIT is a species of FORWARD, and vice-versa, and is also a species of REQUEST, but FORWARD is not a species of REQUEST. To clarify these definitions, one needs to define FORWARD and RECRUIT formally, and perform the compositions with REQUEST. Moreover, only with a formal semantics can one define precisely what is meant by 'being a species of'.

The point of providing a semantics for a composed action is only to illustrate how a semantics such as ours may be helpful in analyzing communicative acts. There is no recommendation here being made that agent designers should use this semantics, or any semantical treatment, for analyzing their systems. Rather, the KQML community needs to know that the language *has* a semantics, and that a strong foundation is present upon which to build.

8 Impact on KQML Design Decisions

Two major decisions have influenced the design of KQML, namely extensibility and content independence. Given that we have questioned the nature of the basic KQML elements (i.e., performatives), what impact will this critique have on those decisions?

[11] Compare with the stipulations of [18].

8.1 Extensibility

By allowing the set of performatives to be extended, KQML requires developers to carry the burden of ensuring that agents know how to behave with these new communication actions. Unfortunately, developers are given no guidance on how to formulate new performatives. Merely providing a new label does not ensure that the act does anything new. Moreover, without a true semantics, it is difficult to rule out incoherent and/or self-defeating definitions.

As the language currently stands, the directive force of most performatives is implicit in the semantics. Therefore, developers need to implement correctly that implicit directive force for each new action of this class. Moreover, those new speech acts need to enter into old and new agent conversation patterns correctly. For example, the designer needs to understand how it is that requests are acknowledged, committed to, discharged, etc. Rather than have to reimplement separate interaction patterns for each of the new actions, the relevant conversation patterns should only be defined once, and then all species of that class should inherit the basic structure.

8.2 Content Independence

In the interest of modularity, KQML handles the communicative actions independently of the content language. This decision can be maintained so long as the content language does not allow any attitude operators (so far, attitude operators are allowed). If BEL, GOAL, INTEND, etc. are allowed in the content language, then agents can easily send self-defeating speech acts to one another, causing undefined behavior. For example, an agent x could TELL a message expressing Moore's paradox, namely, $(\mathsf{p} \wedge \neg(\mathsf{BEL}\ \mathsf{x}\ \mathsf{p}))$. To generalize, an agent could say both that it requests an act, and that it does not want the act done, etc. Thus, agents could deny the sincerity conditions or preparatory conditions [25] of the relevant speech act. Without a semantics, this may not be detectable by an agent designer. It is unlikely, however, that the agent architecture could detect such semantic anomalies automatically.

A disadvantage of content independence is that it prevents the content from being checked for compatibility with the speech act type. For example, directives employ have future actions, and actions that are promised should be only those to be performed by the promising agent.

9 Comparison with other Agent Communication Languages

A number of other agent communication languages have been proposed, including AOP [27, 31] and Telescript [32]. AOP shares with the approach proposed here a concern for agents' commitments as a crucial part of any agent communication language and framework. Although our definitions and semantics of the foundational concepts of choice and intention are different, their importance to both efforts is similar. We differ in that the approache presented here attempts to provide a logical semantics of communicative acts per se, whereas

AOP essentially provides rules for responding to or issuing a message with a speech act label (i.e., REQUEST, INFORM, or QUERY-WHETHER). This operational semantics leaves it open to the agent developer what the communicative acts in fact mean, and how they affect the agents' mental states. We advocate there being an agreement or specification about what the communication acts are supposed to mean, as it may be important for technical, as well as legal reasons, to show that agents behaved according to the specifications.

There are technical differences, of course, in the details of communicative action. For example, because there is no nondeterministic choice operator that can form disjunctive actions in AOP, a yes/no question is defined as three requests to inform about whether the queried proposition is true, false, or unknown. In our model, there is only one request, but for a disjunctive act. The "I don't know" response is handled at a higher level, and need not be specified as part of the semantics of a question.

Telescript seems not to be an agent communication language, per se, but rather a scripting language. Agents do not communicate in the sense of sending messages to one another. Rather, they transport themselves to various "places," at which they "meet" other agents. It is claimed, then, that agents "interact using object-oriented programming techniques." This raises the questions of what the agents' methods are, and how one agent knows what another's are. Let us assume that all agents are specializations of a given class, and share a common set of methods. Then, one might expect that an agent would communicate by executing another agent's public methods rather than execute any communicative acts per se. For example, one agent may execute another's "sell a ticket" method, which would start the process of ticket sales. This type of interaction makes sense when a homogeneous network is involved, in which agents share methods and have knowledge of each other's methods. For example, the ticket buyer does not share a "sell a ticket" method with the seller; rather it must somehow be designed to expect such a method when it reaches the ticket seller. In the heterogeneous world of the internet, it is unclear how such agent-programming would talk place. Moreover, and especially in a business context, we are concerned about the nature of communicative actions in an environment in which some agents are insincere, deceptive, or merely antagonistic. Our semantics is designed to insulate an agent's mental states from being operated upon directly by other agents. On receiving a message, an agent is supposed to have a "model" of what the other agent *wants* it to do/believe, etc. By representing the effects of communicative actions in this way, the receiving agent then has the option to reason about the sincerity of the sender, while still protecting its innermost states. A developer of Telescript agents may in fact be able to afford such protection to its agents, but as specified publicly so far (in [32, this volume]), no guidance is given on how to do this.

9.1 Agent architectures may include people

We began this paper by observing that an agent architecture needs to interact with people. End users will give the architecture tasks to be performed, and in the course of execution the agents may need to make requests of other people. For example, in the OAA [9], the NOTIFY agent may need to have a telephone phone conversation with the person who answers the telephone in order to pass a message to the intended recipient. Thus, to specify a *hybrid* agent architecture, including people and software agents, the semantics will need to make reference to "states" that the human may be in, such as being committed to bringing the intended recipient to the telephone. The agents will need to expect certain behavior of human agents who may take part, perhaps accidentally, in their activities, and they themselves will need to engage in behavior that is at least sufficiently familiar that people can readily adapt to their limitations. Rather than have two sets of semantics, it would be parsimonious to develop one model and language that can scale from simple software agents to even an overly simplistic approximation of people.

Our speech act theory has the virtue that essentially the same analysis can be given for human and software agents' communication acts, though obviously, the mental states and surface linguistic forms inherent in human communication are far richer in the human case. Still, one may argue that very limited user interfaces could be generated based on a much simpler agent model — for example, the Magic Cap[12] interface employs Telescript as its core, which does not have the richness advocated here. Again, this may work for the case of simple devices, with simple functionality, in a homogenous agent environment. But, we are dubious for more complex and heterogenous environments, especially those that might involve telephone interactions. We are not advocating that natural language interaction be the only way users interact with their agents (in fact, we build multimodal interfaces [7, 9, 21]). But, agent architectures will certainly need to be developed that can support telephone-based interaction.

9.2 Future Work: Interagent Dialogues

The communicative action definitions given here are in need of modification and repair, as they are too one-sided. They express the sender's mental states, but are not sufficient to initiate team behavior [13]. The reason is that neither sender nor receiver ends up with the right kinds of commitments that together can establish a joint intention. In [8, 29], we redefine the communicative actions of request, refuse, etc., showing how joint intentions are established, monitored, and discharged. Moreover, we demonstrate how joint intention theory can predict the structure of finite-state models of interagent dialogue, which have been claimed to provide an adequate model of interagent dialogue [4, this volume].

10 Conclusion

KQML has shown its versatility in supporting agent architectures as a viable computational paradigm. However, the language is still in need of a more precise definition and semantics before agent designers can be confident that their contributions rest on solid foundations. In particular, commercial uses of agent architectures will at some point emerge, and it will be important for legal as well as technical reasons to be able to show that agents correctly

[12] Magic Cap is a trademark of the General Magic Corporation

performed according to their specifications and those of the language. We believe a semantics something like the one illustrated here will be necessary as part of this effort.

References

[1] J. F. Allen and C. R. Perrault. Analyzing intention in dialogues. *Artificial Intelligence*, 15(3):143–178, 1980.

[2] D. Appelt. *Planning English Sentences.* Cambridge University Press, Cambridge, U. K., 1985.

[3] J. L. Austin. *How to do things with words.* Oxford University Press, London, 1962.

[4] J. M. Bradshaw, S. Dutfield, P. Benoit, and J. D. Wooley. KAoS: toward an industrial-strength open agent architecture. In J. Bradshaw, editor, *Software Agents.* AAAI/MIT Press, Cambridge, Massachusetts, in press. this volume.

[5] M. Bratman. *Intentions, Plans, and Practical Reason.* Harvard University Press, 1987.

[6] B. C. Bruce. Belief systems and language understanding. Technical Report 2973, Bolt Beranek and Newman Inc., Cambridge, Masschusetts, 1975.

[7] P. R. Cohen. The role of natural language in a multimodal interface. In *The 2nd FRIEND21 International Symposium on Next Generation Human Interface Technologies*, Tokyo, Japan, November 1991. Institute for Personalized Information Environment. Also appears in Proceedings of UIST'92, ACM Press, New York, 1992, 143-149.

[8] P. R. Cohen. Models of dialogue. In M. Nagao, editor, *Cognitive Processing for Vision and Voice: Proceedings of the Fourth NEC Research Symposium.* SIAM, 1993.

[9] P. R. Cohen, A. Cheyer, M. Q Wang, and S. C. Baeg. An open agent architecture. Appears in the Proceedings of the AAAI Spring Symposium Series on Software Agents, 1994.

[10] P. R. Cohen and H. J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42(3), 1990.

[11] P. R. Cohen. and H. J. Levesque. Performatives in a rationally based speech act theory. In *Proceedings of the 28th Annual Meeting, Association for Computational Linguistics*, Pittsburgh, Pennsylvania, 1990.

[12] P. R. Cohen and H. J. Levesque. Rational interaction as the basis for communication. In P. R. Cohen, J. Morgan, and M. E. Pollack, editors, *Intentions in Communication.* MIT Press, Cambridge, Massachusetts, 1990.

[13] P. R. Cohen and H. J. Levesque. Teamwork. *Noûs*, 25(4):487–512, 1991. Also Technical Note 504, Artificial Intelligence Center, SRI International, Menlo Park, California, 1991.

[14] P. R. Cohen and C. R. Perrault. Elements of a plan-based theory of speech acts. *Cognitive Science*, 3(3):177–212, 1979.

[15] ARPA Knowledge Sharing Initiative External Interfaces Working Group. Specification of the KQML agent-communication language. Working paper, 1993.

[16] T. Finin, R. Fritzson, D. McKay, and R. McEntire. KQML as an agent communication language. In *Proceedings of the Third International Conference on Information and Knowledge Management (CIKM'94)*, New York, November 1994. ACM Press.

[17] M. Genesereth and S. Ketchpel. Software agents. *Communications of the ACM*, 37(7):100–105, July 1994.

[18] Y. Labrou and T. Finin. A semantics approach for KQML — a general purpose communication language for software agents. In *Proceedings Proceedings of the Third International Conference on Information and Knowledge Management (CIKM'94)*, Washington, D.C., 1994.

[19] J. McCarthy. ELEPHANT: a programming language based on speech acts. Unpublished ms., Dept. of Computer Science, Stanford University, 1989.

[20] D. McDermott. A temporal logic for reasoning about processes and plans. *Cognitive Science*, 6(2):101–155, April-June 1982.

[21] S. L. Oviatt, P. R. Cohen, and M. Q. Wang. Toward interface design for human language technology: Modality and structure as determinants of linguistic complexity. *Speech Communication*, 15(3-4), 1994.

[22] C. R. Perrault. An application of default logic to speech act theory. In P. R. Cohen, J. Morgan, and M. E. Pollack, editors, *Intentions in Communication.* M.I.T. Press, Cambridge, Massachusetts, 1990.

[23] C. R. Perrault and J. F. Allen. A plan-based analysis of indirect speech acts. *American Journal of Computational Linguistics*, 6(3):167–182, 1980.

[24] D. Sadek. Dialogue acts are rational plans. In *Proceedings of the ESCA/ETRW Workshop on "The structure of multimodal dialogue" (VENACO II)*, Maratea, Italy, September 1991.

[25] J. R. Searle. *Speech acts: An essay in the philosophy of language.* Cambridge University Press, Cambridge, 1969.

[26] J. R. Searle and D. Vanderveken. *Foundations of Illocutionary Logic.* Cambridge Univ. Press, New York City, New York, 1985.

[27] Y. Shoham. Agent-oriented programming. *Artificial Intelligence*, 60(1):51–92, 1993.

[28] C. Sidner. An artifical discourse language for collaborative negotiation. In *Proceedings of the National Conference on Artificial Intelligence (AAAI'94)*, pages 814–819. AAAI Press, 1994.

[29] I. Smith and P. R. Cohen. A semantics for interagent dialogue. in preparation, 1995.

[30] R. G. Smith and R. Davis. The contract net protocol: High-level communication and control in a distributed problem solver. In A. H. Bond and L. Gasser, editors, *Readings in Distributed Artificial Intelligence*, pages 357–366. Morgan Kaufmann, San Mateo, Caifornia, 1988.

[31] B. Thomas. *PLACA, an agent oriented language.* PhD thesis, Department of Computer Science, Stanford University, Stanford, California, 1993.

[32] J. E. White. Mobile agents. In J. Bradshaw, editor, *Software Agents*. AAAI/MIT Press, Cambridge, Massachusetts, in press. this volume.

11 Appendix: Semantics and Model Theory

A model M is a structure $< \Theta, P, E, Agt, T, B, G, \Phi >$, where Θ is a set of things, P is a set of people, E is a set of primitive event types, $Agt \in [E \longrightarrow P]$ specifies the agent of an event, $T \subseteq [Z \longrightarrow E]$ is a set of possible courses of events (or worlds) specified as a function from the integers to elements of E, $B \subseteq T \times P \times Z \times T$ is the belief accessibility relation, $G \subseteq T \times P \times Z \times T$ is the goal accessibility relation, and Φ interprets predicates. Formulas will be evaluated with respect to some possible course of events, hereafter some *possible world*, and an "index" into that possible world.

11.0.1 Definitions

- $D = \Theta \bigcup P \bigcup E^*$, specifying the domain of quantification. Note that quantification over sequences of primitive events is allowed.
- $\Phi \in [Pred^n \times T \longrightarrow 2^{D^n}]$, specifying the interpretation of predicates.
- $AGT \subseteq T \times P$, where $x \in AGT[e_1, \ldots, e_n]$ iff there is an i such that $x = Agt(e_i)$. That is, AGT specifies the partial agents of a sequence of events.

11.0.2 Satisfaction

Assume M is a model, σ a sequence of events, n an integer, v a set of bindings of variables to objects in D, and if $v \in [Vars \longrightarrow D]$, then v^x_d is that function which yields d for x and is the same as v everywhere else. We give the following definition of $\models$:

1. $M, \sigma, v, n \models P(x_1, \ldots, x_n)$ iff $< v(x_1) \ldots v(x_n) > \in \Phi[P, \sigma, n]$. Notice that the interpretation of predicates depends on the world σ and the event index n.

2. $M, \sigma, v, n \models \neg\alpha$ iff $M, \sigma, v, n \not\models \alpha$

3. $M, \sigma, v, n \models (\alpha \vee \beta)$ iff $M, \sigma, v, n \models \alpha$ or $M, \sigma, v, n \models \beta$.

4. $M, \sigma, v, n \models (x_1 = x_2)$ iff $v(x_1) = v(x_2)$.

5. $M, \sigma, v, n \models \exists x \alpha$ iff $M, \sigma, v^x_d, n \models \alpha$ for some d in D.

6. $M, \sigma, v, n \models (\mathsf{AGT}\ x_1\ e_2)$ iff $AGT[v(e_2)] = \{v(x_1)\}$. AGT thus specifies the *only* agent of event e_2.

7. $M, \sigma, v, n \models \langle\text{Time-Proposition}\rangle$ iff $v(\langle\text{Time-Proposition}\rangle) = n$. Time propositions are currently just numerals. However, for ease of exposition, we shall write them as if they were time-date expressions such as 2:30PM/3/6/85. These will be true or false in a course of events at a given index iff the index is the same as that denoted by the time proposition (i.e., numeral). Depending on the problem at hand, we may use timeless propositions, such as (AT ROBOT NY). Other problems are more accurately modeled by conjoining a time proposition, such as (AT ROBOT NY) $\wedge$ 2:30PM/3/6/85. Thus, if the above conjunction were a goal, both conjuncts would have to be true simultaneously.

8. $M, \sigma, v, n \models (\mathsf{BEL\ x}\ \alpha)$ iff for all σ^* such that $\sigma B[v(x)]\sigma^*$, $M, \sigma^*, v, n \models \alpha$. That is, α *follows from* the agents beliefs iff α is true in all possible worlds accessible via B, at index n.

9. $M, \sigma, v, n \models (\mathsf{GOAL\ x}\ \alpha)$ iff for all σ^* such that $\sigma G[v(x)]\sigma^*$, $M, \sigma^*, v, n \models \alpha$. Similarly, α *follows from* the agents goals iff α is true in all possible worlds accessible via G, at index n.

10. $M, \sigma, v, n \models (\mathsf{HAPPENS}\ a)$ iff $\exists m, m \geq n$, such that $M, \sigma, v, n[\![a]\!]m$. That is, a is a sequence of events that happens "next" (after n). $[\![]\!] \subseteq [T \times Z \times D \times ActionExpressions \times Z]$ is an anonymous relation constructor, mutually recursive with $\models$, that relates an action expression to two indices on the course of events. It is described below.

11. $M, \sigma, v, n \models (\mathsf{DONE}\ a)$ iff $\exists m, m \leq n$, such that $M, \sigma, v, m[\![a]\!]n$

    That is, the sequence of events denoted by a has *just* happened, where

12. $M, \sigma, v, n \models (\mathsf{DONE}\ a)$ iff $\exists m, m \leq n$, such that $M, \sigma, v, m[\![a]\!]n$

    That is, the sequence of events denoted by a has *just* happened, where

    $[\![]\!] \subseteq [T \times Z \times D \times ActionExpressions \times Z]$ is characterized by:

Event variables:	$M, \sigma, v, n[\![x]\!]n+m$ iff $v(x) = e_1 e_2 \ldots e_m$ and $\sigma(n+i) = e_i, 1 \leq i \leq m$. Intuitively, x denotes some sequence of events of length m which appears next after n in the world σ.
Null actions:	$M, \sigma, v, n[\![NIL]\!]n$
Alternative actions:	$M, \sigma, v, n[\![a \mid b]\!]\sigma_1$ iff $M, \sigma, v, n[\![a]\!]\sigma_1$, or $M, \sigma, v, n[\![b]\!]\sigma_1$.
Simultaneous actions:	$M, \sigma, v, n[\![a \mid b]\!]\sigma_1$ iff $M, \sigma, v, n[\![a]\!]\sigma_1$, and $M, \sigma, v, n[\![b]\!]\sigma_1$.
Sequential Actions:	$M, \sigma, v, n[\![a;b]\!]m$ iff $\exists k, n \leq k \leq m$, such that $M, \sigma, v, n[\![a]\!]k$ and $M, \sigma, v, k[\![b]\!]m$
Test Actions:	$M, \sigma, v, n[\![\alpha?]\!]n$ iff $M, \sigma, v, n \models \alpha$
Iterative Actions:	$M, \sigma, v, n[\![a^*]\!]m$ iff $\exists n_1, \ldots, n_k$ where $n_1 = n$ and $n_k = m$ and $\forall i, 1 \leq i \leq m$ $M, \sigma, v, n_i[\![a]\!]n_{i+1}$

The test action, $\alpha?$, is an action expression that denotes either the action NIL if α holds, or "blocks" (fails), if α is false. The operators HAPPENS and DONE, when applied to test actions $\alpha?$, essentially constrain the possible worlds to be ones where α is true at the requisite time. For example, (HAPPENS a;(POOR x)?) is true iff a sequence of events described by the action expression a happens next after which x is poor. The iterative action a^* denotes a finite sequence of a's. A wff α is *satisfiable* if there is at least one model M, world σ, index n, and value assignment v such that $M, \sigma, v, n \models \alpha$. A wff α is *valid*, iff for every model M, world σ, event index n, and assignment of variables v, $M, \sigma, v, n \models \alpha$.