In Proceedings of the IEEE International Phoenix Conference on Computers and Communications, 1992

ON THE SEMANTICS OF PROTOCOLS AMONG DISTRIBUTED INTELLIGENT AGENTS

Munindar P. Singh*

Dept of Computer Sciences and OODS Lab
University of Texas MCC
Austin, TX 78712 Austin, TX 78759
USA USA

msingh@cs.utexas.edu

ABSTRACT. The continuing expansion of distributed intelligent systems makes new demands on theories of communication in Computer Science. It is customary to describe the individual nodes or *agents* in an intelligent system in terms of higher level concepts like intentions, know-how and beliefs. However, current theories of the communication among such agents provide no form of a formal or rigorous semantics for the messages exchanged at a corresponding level of abstraction—they either concern themselves with implementational details or address what is, for artificial systems, an irrelevant aspect of the problem. A recent theory of communication that gives the *objective* model-theoretic semantics for speech acts is applied to this problem. This allows the important properties of protocols to be formalized abstractly, i.e., at the level of the application, not of the implementation. Further constraints on "good" designs can also be stated, which simplify the requirements imposed on the member agents. The resulting theory not only provides some insights into designing distributed intelligent systems, but also helps in their validation. As an example, it is applied to a logical reconstruction of the classical Contract Net protocol.

I. INTRODUCTION

The trend towards the development of increasingly intelligent systems is matched only by the trend towards the distribution of computing. Distributed Artificial Intelligence (DAI) lies at the intersection of these trends. Besides the well-known reasons for the usefulness of distributed systems, the continued development of DAI systems is attractive for the following reasons. DAI permits intelligent systems to be developed independently of each other and to be reused as components of new systems, i.e., as member *agents* in multiagent systems. This modularization is useful when expertise is distributed, as in medical diagnosis. It also adds to the robustness of the designed system by simplifying the acquisition and validation of knowledge relevant to different aspects of the domain. Moreover, it simplifies design for applications such as manufacturing planning and air-traffic control by allowing an intelligent agent to be located at the site where the data are available and where decisions have to be taken.

A major bottleneck in the design of DAI systems is the design of the protocols of interaction among their member agents. Unfortunately, while individual agents are usually described in terms of their knowledge, intentions and capabilities (i.e., high-level concepts), extant approaches to understanding the interactions between them are not sufficiently advanced. Even fairly recent DAI research, which provides primitives for communication among agent has tended to be concerned with the workings of the TCP/IP and similar protocols, i.e., it has not been possible to abstract out entirely aspects of communication roughly at or below the so-called Transport Layer of the classical ISO/OSI standard (e.g., see [Arni *et al.*, 1990]). Even more to the point, current theories do not provide any kind of a formal or rigorous semantics for the messages exchanged in a DAI system.

This lack of a rigorous theory of the interactions among agents forces the system designer to think in terms of what are, from the point of view of DAI, merely details of the underlying architecture—these details are important, but are simply out of place in the context of DAI. The resulting mixing up of concerns often results in the behavior of the designed system depending crucially on details of the operating system and the network hardware. At the same time, the design of the individual agents is based on knowledge about the domain of application that they have at different stages of their computations. Thus there is no principled way to relate the interactions *among* the agents to the knowledge *within* each of them. The designer must design some acceptable modes of interaction and relate them as best as possible to the knowledge of the agents. Not only is this a tedious task, it also has to be redone from the start if the system is ever re-implemented. And no help is provided when systems implemented in different ways are to be integrated. In short, the problems with extant technology are that

1. It requires that the interactions among agents be designed from scratch each time.
2. The semantics of these interactions is embedded in the procedures, some of which involve network and operating

*This research was supported by the National Science Foundation through grant # IRI-8945845 to the Center for Cognitive Science, University of Texas, Austin Texas, by DFKI (the German Research Center for Artificial Intelligence), Kaiserslautern, Germany, and by MCC (the Microelectronics and Computer Technology Corporation), Austin, Texas.

system code. This makes the validation and modification of systems, even otherwise not trivial, even more difficult.

3. Systems designed independently cannot be easily integrated.
4. Graceful updation or redesign of a system is virtually impossible: one cannot easily replace an existing agent with a new one.

Taken together, these limitations subvert many of the main motivations for developing DAI. The goal of this paper is to present a theory of the interaction among agents and a formal semantics for their interactions. Our key methodological assumptions are the following. We take it for granted that intelligent agents can be best described (for design or analysis) with concepts such as intentions, know-how or beliefs. This is quite a standard assumption in AI [McCarthy, 1979]. We consider DAI systems from without, i.e., as designers and analyzers. We do *not* directly take the point of view of the different agents who compose the system. Thus we attribute beliefs and intentions to agents, and describe their communications as we see fit from an "external" viewpoint rather than how they might actually be represented in the agents. This is useful since this leaves the exact design of the agents an open issue to be settled later in the design process, provided they meet the minimal requirements imposed.

Recently much work has been done on the design of protocols based on a notion of "knowledge" [Halpern and Moses, 1987]. However, papers on this theme consider the knowledge that the processes have of the process of communication itself, e.g., about whether certain messages have been delivered to the intended recipient or not. Also, these protocols are designed for lower level data transmission. The work reported here is significantly different in that it emphasizes and studies the semantics of the messages exchanged, not the process of exchanging them.

The rest of the paper is organized as follows. In section II., we broadly classify the kinds of communicative interaction that occur most often in DAI systems, briefly describe *Speech Act Theory* and relate it to those interactions. In section III., we describe a recent formal theory of the objective semantics of the major kinds of speech acts. In section IV., we show how this theory can be applied to the understanding of protocols in DAI systems. In section V., we present a detailed example of the logical reconstruction of the *Contract Net*, a celebrated protocol in DAI, which we also describe within.

II. KINDS OF INTERACTIONS AMONG AGENTS

The behavior of a DAI system depends not just on its component agents, but also on how they interact. In the more interesting cases, the agents would also intelligently decide how to interact with other agents by considering their own current situation at that time.

II–A. Protocols

Therefore, in a DAI system of sufficient complexity, each agent would not only need to be able to do the tasks that arise locally, but would also need to interact effectively with other agents. We take *protocols* to be the specifications of these interactions. Agents participate in different protocols by appropriately interacting with each other, e.g., by responding to messages, performing actions in their given domain, or updating their local states. Protocols can thus be taken as specifying the *policies* that the agents would follow with regard to their interactions with other agents. These policies would, e.g., determine the conditions under which a request would be acceded to or a permission issued or a statement believed. These policies could be fixed to some extent at the time of design, but would involve significant components that depended on the agents' current situation and thus could be computed only during execution; e.g., a request might be acceded to only if acceding to it does not lead to task overload. Protocols, when seen in this way, are a nice way of enforcing modularity in the design of a DAI system. They help in separating the interface between agents from their internal design. These protocols are meant to be rather high-level; in the classical seven-layer ISO/OSI framework, they would lie in the application layer. Some of these protocols may, in practice, precede "real" applications-level communication by facilitating the setting up of another protocol. This distinction is not crucial for our purposes.

Several kinds of formalizations may be attempted for protocols. One kind would concern the deliberation processes of the agents as they decide how to respond to a message. These processes are highly nonmonotonic and can be accurately understood only with theories of belief and intention revision, which are still not sufficiently well-developed (e.g., see [Perrault, 1987]). Another formalization concerns the objective conditions of satisfaction for different kinds of messages. This is the one attempted here. Not only is this useful from the point of view of design, it also helps clarify our intuitions about the process of deliberation involved since ideally the agents should act so as to "satisfy" the messages communicated in their system. We return to this point in section VI..

II–B. Speech Act Theory

Speech Act Theory deals with natural language utterances. Initially, it was developed to deal with utterances, e.g., "I declare you man and wife," that are not easily classified as being true or false, but rather are actions. Later it was extended to deal with all utterances, with the primary understanding that all utterances are *actions* of some sort or the other [Austin, 1962; Bach and Harnish, 1979; Searle, 1969]. A speech act is associated with at least three distinct actions: (1) a *locution*, i.e., the corresponding physical utterance, (2) an *illocution*, i.e., the conveying of the speaker's intent to the hearer, and (3) any number of *perlocutions*, i.e., actions that occur as a result of the illocution. For example, "shut the door" is a locution, which might be the illocution of a command to shut the door, and might lead to the perlocution of the listener getting up to shut the door. All locutions do not also count as illocutions—some of them may be just occur in the wrong situation. All perlocutions are not caused by appropriate illocutions—some of them may occur because of other contextual features. For this reason, A speech act *per se* is usually identified with its associated illocution [Searle, 1969]. We adopt this practice in this paper.

Speech acts may be classified into a small number of interesting classes, including *assertives, directives, commissives, permissives* and *prohibitives*. Briefly, assertives are statements of fact; directives are commands, requests or advice; commissives (e.g., promises) commit the speaker to a course of action; permissives issue permissions; and prohibitives take them away [Singh, 1991d]. These classes are said to have different *illocutionary forces:* they can be combined with the same *proposition* to yield different illocutions; e.g., "the door is shut" is an assertive and "shut the door" a directive, both of which apply to the same proposition, namely, that the door is shut—the assertive says that this proposition *is* true; the directive asks that it be *made* true [Searle, 1969].

II–C. Speech Act Theory in DAI

Speech Act Theory has also been found useful in DAI as a foundation for communication among agents. We agree with this view. There are two kinds of applications of Speech Act Theory in DAI. The first, and by far the more common one, uses it to motivate different *message types* for interactions among agents. The idea is that since agents can perform different kinds of speech acts, the language used for communication must allow different types of messages [Huhns *et al.*, 1990; Thomas *et al.*, 1990]. This is quite standard, and something we shall do ourselves. However, these proposals are informal—they rely on ones understanding of the labels used to understand the meanings of the different message types. The true semantics is embedded in the procedures that manipulate different messages.

The second kind of application of Speech Act Theory in DAI yields more sophisticated theories, which treat illocutions as linguistic actions and aim to describe the interactions of agents in terms of what they say to each other. These theories attempt to generalize linguistic theories of communication designed for human communication to the domain of DAI [Cohen and Levesque, 1988]. As a result, they tend to be somewhat top-heavy; e.g., they require that each of the agents involved have beliefs about the others' beliefs about their beliefs, and so on, *ad infinitum.* It is known that such *mutual beliefs* are not achievable in asynchronous systems [Fischer and Immerman, 1986; Halpern and Moses, 1987]. But more to the point, these theories suffer from being based on traditional formalizations of speech acts [Allen and Perrault, 1980]. Traditional formalizations are primarily concerned with identifying different kinds of illocutions. Thus these theories give the conditions under which saying "can you pass the salt?" is not a question, but rather a request; it is then an *indirect* speech act. An example of a condition for requests might be that the speaker and hearer mutually believe that the speaker has certain intentions and beliefs. The phenomenon of indirect speech acts is, no doubt, of great importance in understanding natural language. But it is of simply no use in any conceivable DAI system: DAI systems can function quite well with just an artificial language that can be simply designed to be free of the ambiguities that these theories have been created to detect.

In a DAI scenario, we can have agents specify explicitly whether they intend their communication to be a request or a promise or an assertion or whatever. Thus the interesting part of the semantics of speech acts, as they may be applied in DAI, concerns what they cause to be done rather than whether they are interpreted to be of one kind or another. At least as a first approximation, we can assume that the illocutionary force of a message transmitted be just the one that is obvious from its syntax. Thus we will not consider indirect speech acts, whose primary role in human language seems to be to permit communication that in the direct form might be culturally unacceptable.

III. FORMAL SEMANTICS FOR COMMUNICATION

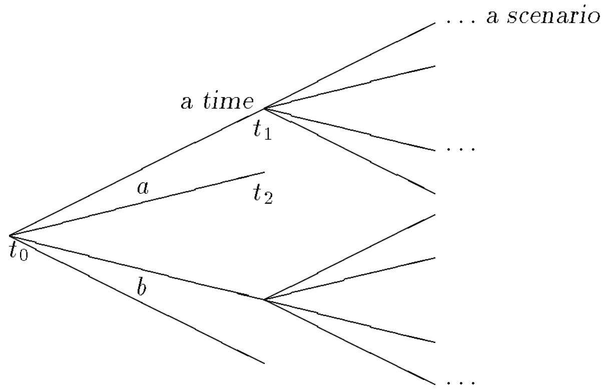

Figure 1: A World with a Branching History

The formal model of this theory posits a set of possible worlds. As diagramed in Figure 1, each possible world is in one of several states, and may develop in any of several ways depending on the agents' actions and, possibly, other events; e.g., the state of the world may change from t_0 to t_1 or to t_2, if the given agent does action a, depending on what else happens at that time. Each of the different ways in which a world may develop is called a *scenario* and is equivalent to a possible course of events. A stretch of time on a scenario, S, from time, t, to time, t', is called a *subscenario*, and is notated as $\langle S, t, t' \rangle$. The set of scenarios that begin at time t in world w is called $\mathbf{S}_{w,t}$.

Formally, a model, M, is a tuple, $\langle \mathbf{W}, \mathbf{T}, <, \mathbf{A}, [\![\,]\!] \rangle$, where $\mathbf{W}$ is a set of possible worlds, $\mathbf{T}$ is a set of times composing them, and $\mathbf{A}$ is a set of agents. The partial order of times is captured by the relation, $<$. $[\![\,]\!]$ gives the set of worlds-time pairs at which atomic propositions are true and the sets of subscenarios over which actions are done (for actions, $[\![\,]\!]^x$ means that the action was done by agent x). For simplicity, only the communicative actions of agents as described below are considered explicitly in the formal language. Other actions, however, are needed in the model to give a semantics for speech acts and of intentions and know-how, and to accurately describe the given domain. Also, the symbols denoting agents stand for themselves.

Using the idea described at the end of section II–B., we can consider messages as having a simple abstract syntax. A message, m, is a pair $\langle i, p \rangle$, where i identifies the illocutionary force, and p the proposition. Here i is an atomic symbol from the set {directive, commissive, permissive, prohibitive, assertive}; and p is a logical formula. Let 'comm' be a predicate that applies to two agents, and a message. 'Comm(x, y, m)' is true relative to a scenario and a time-point iff message m is uttered to agent y by agent x at that time on that scenario. Let 'says-to(y, m)' be the (only) action that agent x can perform to make 'comm(x, y, m)' true. The time at which this action is completed is important in the semantics.

This syntax allows us to ignore details of message transmission and to focus on the *objective semantics* of speech acts. The idea of an objective semantics for speech acts has been introduced and defended in previous work [Singh, 1991d]. It considers, not the conditions under which a particular kind of speech act may be said to have occurred, but rather the conditions under which it may be said to have been *satisfied* objectively. A transmitted message may not always be satisfiable, e.g., a request may not be granted. In order to be able to talk of the satisfaction of messages explicitly, we introduce an operator WSAT that applies on formulas of the form 'comm(x, y, m)' and states that the corresponding message is *whole-heartedly* satisfied. Conditions of truth may be stated for WSAT applied to any kind of message, relative to a scenario and a time [Singh, 1991d].

The major classes of speech acts have been formalized in this way. As an example, a directive uttered by one agent to another is said to be satisfied along any course of events in which it becomes true, but in such a way that the listener intended it to become true and knew how to make it true; e.g., the directive "shut the door" would be satisfied if the door gets shut eventually, and until it is shut, the listener continuously intends to shut it and knows how to shut it (see Figure 2). The mere shutting of the door is not sufficient, since it could have happened by accident. The concepts of "intention" and "know-how" as used in this definition have themselves been formalized in the same model of action and time [Singh, 1991a; Singh, 1991b; Singh, 1991c]. The details of those formalizations are too complex to be included here; however, that they are available is reason to be reassured that the crucial concepts are not undefined.

The formal language of this paper, $\mathcal{L}$, is CTL* (a propositional branching time logic [Emerson, 1989]) augmented with predicates for intention and know-how, and the operator WSAT. Let Φ be a set of atomic propositional symbols. A formula of $\mathcal{L}$ can be any of the following: an atomic formula ($\psi \in \Phi$), a conjunction of formulae ($p \wedge q$), a negation of a formula ($\neg p$), a predicate applied to some arguments, or a scenario-quantifier followed by a scenario-formula. A scenario-quantifier is one of A and E. A denotes "in *all* scenarios at the present time," and $\mathsf{E}p \equiv \neg\mathsf{A}\neg p$. A scenario-formula is an ordinary formula or an until-expression ($p\mathsf{U}q$). $p\mathsf{U}q$ denotes "q holds sometimes in the future on this scenario, and p holds at all times from now until then." $\mathsf{F}p$ denotes "p holds sometimes in the future on this scenario" and abbreviates "$\mathsf{true}\mathsf{U}p$." Note that $p \rightarrow \mathsf{F}p$. $\mathsf{G}p$ denotes "p always holds in the future on this scenario" and abbreviates "$\neg\mathsf{F}\neg p$." $\mathsf{P}p$ denotes "p holds somewhere in the past." Implication ($p \rightarrow q$) and disjunctions of formulae ($p \vee q$) are defined as the usual abbreviations.

The semantics of formulae in $\mathcal{L}$ are given relative to a model as defined above and a world and time in it. $M \models_{w,t} p$ expresses "M satisfies p at w, t." $M \models_{S,t} p$ expresses "M satisfies p at time t on scenario S," and is needed for some formulae. As remarked above, the semantics of know-how and intention are not included here: briefly, intends(x, p), $K_{how}(x, p)$, and $K_{prev}(x, p)$ mean that the agent x, respectively, intends, knows how to achieve, and knows how to prevent the condition p.

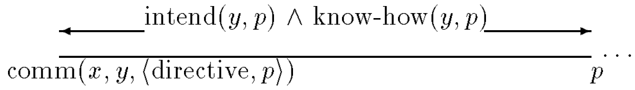

Figure 2: The Satisfaction Condition for Directives

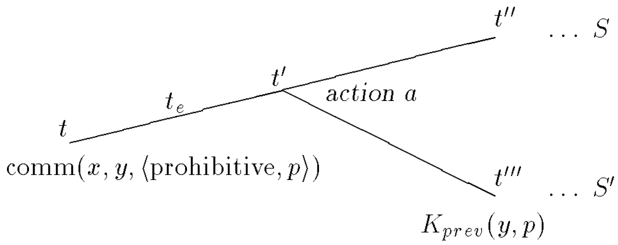

Figure 3: The Satisfaction Condition for Prohibitives

1. $M \models_{w,t} \psi$, for $\psi \in \Phi$ iff $\langle w, t\rangle \in [\![\psi]\!]$
2. $M \models_{w,t} p \wedge q$ iff $M \models_{w,t} p \wedge M \models_{w,t} q$
3. $M \models_{w,t} \neg p$ iff $M \not\models_{w,t} p$
4. $M \models_{S,t} p \wedge q$ iff $M \models_{S,t} p \wedge M \models_{S,t} q$
5. $M \models_{S,t} \neg p$ iff $M \not\models_{S,t} p$
6. $M \models_{w,t} \mathsf{A}p$ iff $(\forall S : S \in \mathbf{S}_{w,t} \rightarrow M \models_{S,t} p)$
7. $M \models_{S,t} p\mathsf{U}q$ iff $(\exists t' : M \models_{S,t'} q \wedge (\forall t'' : t \leq t'' \leq t' \rightarrow M \models_{S,t''} p))$
8. $M \models_{S,t} \mathsf{P}p$ iff $(\exists t' : t' < t \wedge M \models_{S,t'} p)$
9. $M \models_{S,t} p$ iff $M \models_{w,t} p$, if p is an ordinary formula, and w is the (unique) world such that $S \in \mathbf{S}_{w,t}$
10. $M \models_{S,t}$ WSAT(comm$(x, y, \langle$assertive, $p\rangle)$) iff $(\exists t_e : t_e \in S \wedge \langle S, t, t_e\rangle \in [\![$says-to$(x, y, \langle$assertive, $p\rangle)]\!]^x \wedge M \models_{S,t_e} p)$
    An assertive is satisfied iff it holds at the time it is uttered.
11. $M \models_{S,t}$ WSAT(comm$(x, y, \langle$directive$, p\rangle)$) iff $(\exists t_e : t_e \in S \wedge \langle S, t, t_e\rangle \in [\![$says-to$(x, y, \langle$directive, $p\rangle)]\!]^x \wedge (\exists t' \in S : t' \geq t_e \wedge M \models_{S,t'} p \wedge (\forall t'' : t_e \leq t'' < t' \rightarrow M \models_{S,t''} K_{how}(y, p) \wedge$ intends$(y, p))))$
    A directive is satisfied iff the hearer achieves the given condition in the future, and until it does so, it intends to and knows how to achieve it. Please see Figure 2.
12. $M \models_{S,t}$ WSAT(comm$(x, y, \langle$commissive$, p\rangle)$) iff $(\exists t_e : t_e \in S \wedge \langle S, t, t_e\rangle \in [\![$says-to$(x, y, \langle$commissive, $p\rangle)]\!]^x \wedge (\exists t' \in S : t' \geq t_e \wedge M \models_{S,t'} p \wedge (\forall t'' : t_e \leq t'' < t' \rightarrow M \models_{S,t''} K_{how}(x, p) \wedge$ intends$(x, p))))$
    A commissive is satisfied iff the speaker achieves the given condition in the future, and until it does so, it intends to and knows how to achieve it.

13. $M \models_{S,t}$ $\mathsf{WSAT}(\text{comm}(x, y, \langle\text{permissive}, p\rangle))$ iff $(\exists t_e : t_e \in S \wedge \langle S, t, t_e\rangle \in [\![\text{says-to}(x, y, \langle\text{permissive}, p\rangle)]\!]^x \wedge (\exists t' \in S : t' \geq t_e \wedge (\forall a : (\exists t'' : \langle S, t', t''\rangle \in [\![a]\!]^y) \rightarrow (\exists S', t''' : t''' \in S' \wedge \langle S', t', t'''\rangle \in [\![a]\!]^y \wedge M \not\models_{S',t'''} K_{prev}(y, p)))))$

    A permissive is satisfied iff the hearer does some action that could lead to a point from where the hearer would not be able to prevent the given condition. That is, the hearer can risk letting the permitted condition occur.

14. $M \models_{S,t}$ $\mathsf{WSAT}(\text{comm}(x, y, \langle\text{prohibitive}, p\rangle))$ iff $(\exists t_e : t_e \in S \wedge \langle S, t, t_e\rangle \in [\![\text{says-to}(x, y, \langle\text{prohibitive}, p\rangle)]\!]^x \wedge (\forall t' \in S : t' > t_e \rightarrow (\forall a : (\exists t'' : \langle S, t', t''\rangle \in [\![a]\!]^y) \rightarrow (\forall S', t''' : t''' \in S' \wedge \langle S', t', t'''\rangle \in [\![a]\!]^y \rightarrow M \models_{S',t'''} K_{prev}(y, p)))))$

    A prohibitive is satisfied iff the hearer does not do any action that could lead to a point from where the hearer would not be able to prevent the given condition. That is, the hearer cannot even risk letting the prohibited condition occur. Please see Figure 3.

IV. APPLYING THE THEORY

The ways in which a theory of the semantics of speech acts, such as the one used here, may be applied in DAI are perhaps obvious. Such a theory can lead to a clearer understanding of the issues involved in the functioning of DAI systems and can be used in both their design and analysis. The formal model it supplies can be used to verify that a given design has the desired properties. When a given system does not work as expected, this may be traced to a failure in whole-heartedly satisfying some message that should have been so satisfied. A designer may use the semantics of speech acts by restricting the design to be such that it allows only correct scenarios to become actual. Thus the agents must act so that all messages exchanged in certain conditions be satisfied as time passes. For example, in cooperative systems all requests that are "reasonable" (in an appropriate sense, given the system at hand) ought to be acceded to. Similarly, all assertions ought to be true and all promises ought to be kept.

We would like that the design of a DAI system be such that only those scenarios be potentially actualized in it that are in some sense "good" or correct. An obvious requirement for correctness in our framework is that all the messages that arise on a given scenario be WSAT on it. In other words, the design should be constrained such that only those messages occur in it whose satisfaction can be guaranteed by it. There are two ways that a designer might go about enforcing these constraints on the design. One is to increase the capabilities of the agents appropriately, e.g., to increase the know-how of the agents involved so that directives are more easily satisfied, to improve their perceptual and reasoning abilities so that their assertives may be true, or to limit what they may intend in different conditions so that their directives and commissives are achievable. The other approach is to treat messages, e.g., commissives, as setting up commitments that are later enforced, and limiting directives so that they occur only when a corresponding commitment has been made.

Once these design decisions have been made they can be stated declaratively in our formal language. One can then simply use standard methods in creating or testing designs of distributed intelligent systems. Such methods, which have already been developed for standard temporal logics include checking the satisfiability of sets of formulas (for us, constraints on the design) and for checking whether a given design satisfies a set of constraints (this is called *model checking*). These methods are described in [Emerson, 1989]. For the particular logic of this paper, such automated methods are not yet available. We now give some examples of formalizations of design constraints. It is by no means suggested that all these constraints make sense in all situations—they are stated below merely to exhibit the power of our theory. In the next section, we discuss an extended example that shows how constraints such as these may be used in DAI.

It should be clarified that the propositions used in the messages are descriptions of conditions, of the world, or of the agent's internal states. That is, they include information about the objects and agents that they involve. The exact predicates and objects involved depend on the domain on which our theory is being applied. For example, the proposition "in(elevator, John)" differs from "in(elevator, Bill)." Thus there is no logical contradiction in Bill's not intending that John ride the elevator, while at the same time intending to ride it himself—in fact, if the elevator can hold only one of them, this might seem quite reasonable. The propositions are evaluated at times in the model, and may have different truth values at different such times. The time of reference (e.g., "6:00 pm") could also be worked into a proposition, though this is not attempted here. Another important point is that constraints as stated involve objective conditions, rather than the beliefs of the agents. This is simply because of the normative force of these constraints. For the agents to act appropriately, they would also need to have the relevant beliefs at the relevant times—this too is something that the designer must ensure, if the designed system is to function as desired.

1. **Intending Ones Directives:**

    The proposition of a directive should be intended by its issuer. For example, if an agent requests another agent to raise a certain voltage (in a system they are jointly controlling), this constraint would require that the first agent should intend that the said voltage be raised.

    $\text{comm}(x, y, \langle\text{directive}, p\rangle) \rightarrow \text{intends}(x, p)$

2. **Preference for Local Action:**

    If an agent knows how to achieve a proposition by itself, it should not issue it as a directive. For example, an agent who needs to raise the voltage on a part of a power network it jointly controls with another agent should do so by itself rather than request the other agent. This constraint is especially useful when communication introduces substantial delays or is expensive. In practice, this constraint would have to be limited to apply not just when the given agent knows how to achieve the required condition, but knows how to do it, even if it carries out the actions that it has to do to fulfill other commitments. Thus an agent may request another agent to do a task that it would have done itself, had it not been swamped with other tasks.

    $K_{how}(x, p) \rightarrow \neg\text{comm}(x, y, \langle\text{directive}, p\rangle)$

3. **Weak Consistency for Directives:**

A directive issued by an agent should not clash with the agent's own intentions; i.e., at least in some scenarios, the speaker's intentions and his directives should be compatible. For example, if the agent intends that the voltage V_1 decrease, then it should not even request another agent to raise voltage V_2, if doing so would necessarily raise V_1 as well, i.e., if the satisfaction of the request by the other agent would preclude this agent from carrying out its own intentions. This constraint differs significantly from constraint 1. Constraint 1 says that the issuer intends the given directive; this constraint says that all of the issuer's intentions are consistent with the directive.

$\text{intends}(x, q) \wedge \text{comm}(x, y, \langle\text{directive}, p\rangle) \rightarrow$
$\mathsf{E}[\mathsf{WSAT}\text{comm}(x, y, \langle\text{directive}, p\rangle) \wedge \mathsf{F}q]$

4. **No Loss of Know-How for Issuers of Directives:**

A directive issued by an agent should not clash with the issuer's own intentions and its satisfaction should not reduce the issuer's ability to achieve its intentions. That is, on all scenarios on which the directive is satisfied, the speaker eventually knows how to achieve its intentions—in fact, it is possible for the know-how of the issuer to have increased as a result of the satisfaction of the issued directive. For example, if an agent intends that the voltage, V_1, decrease and requests another agent to raise voltage V_2, then on all scenarios on which this request is whole-heartedly satisfied, the issuer would eventually be able to lower voltage V_1. This could be so because the agent already knew how to lower V_1 and this know-how was preserved, or because the actions of the other agent made it possible for the agent to acquire the relevant know-how.

$\text{intends}(x, q) \wedge \text{comm}(x, y, \langle\text{directive}, p\rangle) \rightarrow$
$\mathsf{A}[\mathsf{WSAT}\text{comm}(x, y, \langle\text{directive}, p\rangle) \rightarrow \mathsf{F}K_{how}(x, q)]$

5. **Weak Consistency for Prohibitives:**

A prohibitive is issued by an agent only if the agent itself does not intend that it be violated. That is, the agent who prohibits another from letting a certain condition occur should not itself try to make it happen. This is a minimal level of cooperation or rationality one expects from the issuers of prohibitions. For example, if an agent prohibits another agent from connecting to a certain power outlet, he could not be intending that the latter connect to it. Recall the discussion in this section on the nature of propositions. Note that (the agent who prohibited the other from connecting to an outlet) might itself intend to connect to that outlet—but this is a different proposition.

$\text{comm}(x, y, \langle\text{prohibitive}, p\rangle) \rightarrow \neg\text{intends}(x, p)$

6. **Weak Consistency for Permissives:**

A permissive is issued by an agent only if the agent itself does not intend that the relevant proposition never occur. That is, the agent who permits another from letting a certain condition occur should not itself intend to prevent it from ever occurring. This is required so that permissives are issued only felicitously—if an agent does not intend that a given condition ever hold, then it should not permit others to let it hold. For example, if an agent intends to keep a certain power outlet available for its own use, it should not permit others to use it, because that could only render it unavailable at certain times in the future.

$\text{comm}(x, y, \langle\text{permissive}, p\rangle) \rightarrow \neg\text{intends}(x, \neg\mathsf{AG}p)$

Certain potential counterexamples to the applicability of this constraint may be resolved as involving more complex propositions. One case involves game playing, where an agent seemingly permits another to beat it, but intends to win nonetheless. While this constraint is meant only as an example and need not apply in all cases, in this particular case, the permissive may be seen as merely being for playing, i.e., for *trying* to beat the issuing agent. The actions of the hearer could, on some scenarios, lead to the speaker being beaten, but the speaker would prevent such scenarios from becoming actual. Once a game begins, the two agents are peers and neither can permit or prohibit the other.

7. **Consistency of Directives and Prohibitives:**

An agent must not issue a directive and a prohibitive for the same condition, even to two different agents. That is, there is never a scenario on which such a directive and a prohibitive occur. This is a requirement of felicitous communication, since it prevents the speaker from playing off two agents against one another. For example, if an agent directs an agent to (take actions to) raise voltage V_1, it should not require another agent to prevent that very condition. The latter's success essentially precludes the former from succeeding with the directive.

$\neg\mathsf{E}[\mathsf{F}\text{comm}(x, y, \langle\text{directive}, p\rangle) \wedge$
$\mathsf{F}\text{comm}(x, z, \langle\text{prohibitive}, p\rangle)]$

Note, however, that the corresponding constraint for permissives and prohibitives might be counterproductive—in some cases, it would be a good idea to violate it. For example, if agent y cannot achieve condition q (say, that the current, I_1, is 500 Amp) for fear of letting V_1 go above 440 V, then a controller x may ask another agent, z to ensure that V_1 stays below 440 V, while permitting y to let it rise. This allows y to do the required action, while preventing the harmful condition of V_1 going above 440 V. This works since permissives only allow certain conditions to be risked: they do not require them to occur.

8. **Prior Commitment:**

A directive should be issued only after a conditional promise is given by the intended receiver that it would obey it. This solves for the issuer the problem of issuing only those directives that would be satisfied, provided the condition that promises are kept is enforced by the design. However, this condition is easier to enforce in a multiagent system, since it depends to a large extent on the actions, know-how and intentions of one agent (the issuer of the promise), rather than on those of several of them. For example, in a banking situation, an agent may request a loan only from the bank that had given him a pre-approved line of credit. For the commissive to be satisfied, p must hold at least once in the future of the directive being uttered by x.

$\text{comm}(x, y, \langle\text{directive}, p\rangle) \rightarrow \mathsf{P}[\text{comm}(y, x, \langle\text{commissive},$
$\mathsf{P}\text{comm}(x, y, \langle\text{directive}, p\rangle) \rightarrow \mathsf{F}p\rangle)]$

V. EXAMPLE PROTOCOLS

V–A. The Contract Net

The Contract Net is among the most well-known and significant protocols in DAI [Davis and Smith, 1983]. While there are several variations possible, in its most basic form it may be described as in Figure 4. We are given a system with several agents. One of them has a task that it has to perform. It cannot do the task entirely locally and splits it into a number of subtasks. Let us consider one of the subtasks that cannot be performed locally. The agent now takes on the role of the *manager*. It sends out a *call for bids* to a subset of the other agents, describing the relevant subtask. Of the other agents, the ones who can and are willing to perform the advertized subtask respond by sending a *bid* to the manager. The manager evaluates the bids received, and selects one of them. It then sends a message *assigning* the subtask to that agent, who then becomes the *contractor*. The contractor performs the assigned task, possibly invoking other agents in the process. Finally, it communicates the *result* of performing the assigned task to the manager. The manager collects the results of all the subtasks of its original task and thus computes its result. If that task was assigned to it by some other agent, it then sends the result to it.

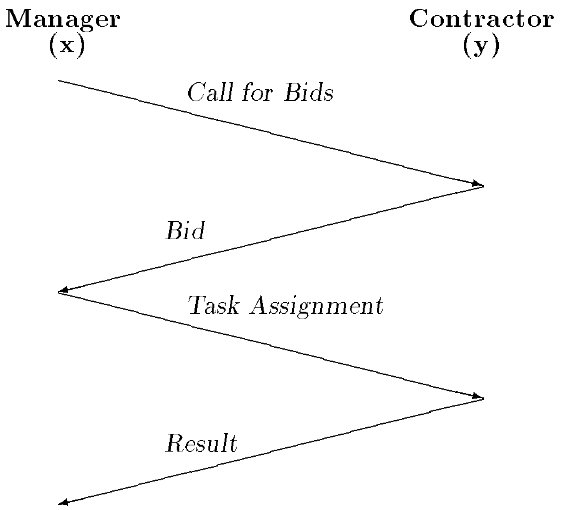

Figure 4: Messages Exchanged in the Contract Net

The key steps in the contract net protocol from our point of view are the following: (1) the call for bids, (2) the bids, (3) the assignment of the task, and (4) the result of the task. The processes of deciding whether to bid on a task and for evaluating the bids when they arrive can be safely abstracted out. These and other steps are local to each agent and involve knowledge of the domain in which the contract net is being used. We assume here that these processes, howsoever designed and implemented, are available and are correct.

One can see almost instantaneously that the message with the result of the task should be classified as an assertive, because, in effect, it states that "the result is such and such." The message making the task assignment is a directive, since it asks the contractor to "do the task!" The message making the bid is a commissive, since it has the force of a conditional promise: "if asked to do the task, I will do it." Finally, the call for bids may itself be treated as a directive, because it has the effect of a request: "please speak up, if you will do this task."

This leads directly to an analysis in which these messages are nested, with the first one to occur being the outermost. Let $\chi(x, y, T)$ capture the conditions under which an agent y will respond to a call for bids sent by x for task, T. Let $r(x, y, T)$ abbreviate $\text{comm}(y, x, \langle \text{assertive}, \text{result}(T)\rangle)$ (*result*); let $a(x, y, T)$ abbreviate $\text{comm}(x, y, \langle \text{directive}, r(x, y, T)\rangle)$ (*assignment*); and let $b(x, y, T)$ abbreviate $\text{comm}(y, x, \langle \text{commissive}, \mathsf{P}a(x, y, T) \rightarrow \mathsf{F}r(x, y, T)\rangle)$ (*bid*).

The initial call for bids has the force of the following schematic message being sent to each of a set of (potential) contractors. The correct performance of the system requires that each instance of this message schema be satisfied by it. Some are satisfied vacuously, if $\chi(x, y, T)$ is false.

- $\langle \text{directive}, \chi(x, y, T) \rightarrow b(x, y, T)\rangle$

In other words, the call for bids is a directive asking the hearer to commit to sending the manager the result of the task, if the manager asks it to send it the result. The assertive with the result of the task is satisfied only if the contractor produces the right result. The contractor must commit to producing the result, if assigned the task (the task can be assigned by sending a simpler message than in the above formalization by taking advantage of the context of communication, but it would logically have the same force as above). Thus the task assignment directive is satisfied if the contractor produces the result when asked to. The call for bids is satisfied if the contractor makes the bid, provided it can perform the given task. As an aside, note that a contractor should not bid on two or more tasks it cannot achieve on some scenarios, i.e., tasks like going North and South simultaneously.

Given that the underlying heuristics, e.g., for selecting one of the bidders, are correct, the above formalization of the contract net can be used to show that it works, if some additional assumptions are made (here x and T are fixed):

- At least one of the agents bids on the task, i.e., $(\exists y : \chi(x, y, T) \mathsf{U} b(x, y, T))$. This means that at least one of the agents is willing and able to perform task T.
- Of the agents who bid on a task, at least one is selected by the manager to award the task to, i.e., $\bigwedge_{1 \leq i \leq n} b(x, y_i, T) \rightarrow (\exists j : 1 \leq j \leq n \wedge a(x, y_j, T))$. This means that at least one of the bidders meets the manager's criteria for task assignment.

The contract net protocol has been designed the way it has been because of some principles of good design. Since the agents involved have limited knowledge about one another, the only way in which the manager can send a given task to the right contractor (short of assigning the task to every available agent), is by first making an utterance that leads to an utterance that restricts the scenarios that can be actualized to those on which the task assignment is guaranteed to be successful. This justifies the sending of the call for bids before making a task assignment and is the canonical motivation for the constraint called *Prior Commitment*, which was introduced in the previous section.

V-B. An Airline Reservation Protocol

The theory of this paper can also be applied in general applications, provided that we can cast them as involving the kinds of messages described here. This can be done for a number of cases. As an example, when someone talks to a travel agent, he can be thought of as requesting the agent to book him on a flight. The agent in turn requests the airline

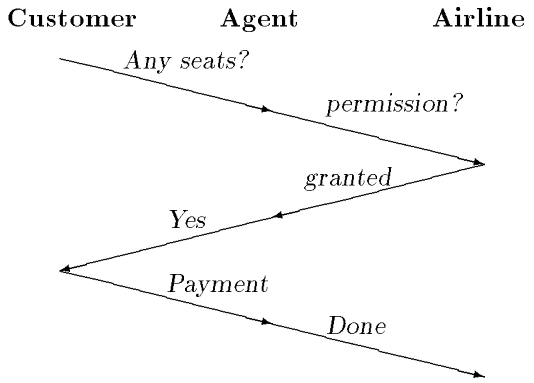

Figure 5: A Protocol for Airline Reservation

(i.e., its reservation system) to grant him permission to sell a ticket, i.e., to commit the airline to fly the pasenger. When the airline does so, the agent promises to commit the airline, if the customer would pay for the ticket. The payment then serves to commit the airline. Lastly, the agent informs the airline that the deal was closed. Arguments similar to the contract net case can be made for this protocol as well.

VI. CONCLUSIONS

Though it is different from previous work on communication in DAI, our approach is compatible with, and complementary to, it. The main difference is that we stress the objective semantics of messages as their most important aspect for DAI. Indeed, if in some system the language of communication cannot be constrained as we have assumed, it might be beneficial to use the traditional theories in determining the truth of $\mathrm{comm}(x, y, m)$, i.e., in computing the illocutionary force of m. Our theory could then be applied at this stage.

We have considered only a few major classes of messages. As more refined categories of messages are considered, we will be able to determine their objective semantics with greater precision, and to specify stronger and, therefore, more useful constraints involving them. We believe that the theory presented in this paper is a first, but important, step in developing a semantics for communication in DAI systems that would yield a rigorous foundation for their design and validation. Eventually formal methods, well-known in temporal logic as used in the validation and design of standard distributed systems may be extended to apply to distributed intelligent systems as well.

VII. ACKNOWLEDGEMENTS

I am indebted to Nicholas Asher, Jürgen Müller and Achim Schupeta for their comments on previous versions of this paper. Talks derived from this paper were given at DFKI (the German Research Center for Artificial Intelligence), Kaiserslautern, Germany, at the University of Nuremberg, Erlangen, Germany, and at the International Computer Science Institute, Berkeley, California. I am indebted to the respective audiences for their comments and discussions.

REFERENCES

Allen, James F. and Perrault, C. Raymond 1980. Analyzing intention in utterances. *Artificial Intelligence* 15:143–178.

Arni, Natraj *et al.*, 1990. Overview of RAD: A hybrid and distributed reasoning tool. Technical Report ACT-RA-098-90, Microelectronics and Computer Technology Corporation, AI Laboratory, Austin, TX.

Austin, John L. 1962. *How to do Things with Words.* Clarendon, Oxford, UK.

Bach, Kent and Harnish, Robert M. 1979. *Linguistic Communication and Speech Acts.* MIT Press, Cambridge, MA.

Cohen, Philip R. and Levesque, Hector J. 1988. Rational interaction as the basis for communication. Technical Report 433, SRI International, Menlo Park, CA.

Davis, Randall and Smith, Reid G. 1983. Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence* 20:63–109. Reprinted in *Readings in Distributed Artificial Intelligence*, A. H. Bond and L. Gasser, eds., Morgan Kaufmann, 1988.

Emerson, E. A. 1989. Temporal and modal logic. In Leeuwen, J.van, editor, *Handbook of Theoretical Computer Science.* North-Holland Publishing Company, Amsterdam, The Netherlands.

Fischer, Michael J. and Immerman, Neil 1986. Foundations of knowledge for distributed systems. In Halpern, Joseph Y., editor, *Theoretical Aspects of Reasoning About Knowledge.* 171–185.

Halpern, Joseph Y. and Moses, Yoram O. 1987. Knowledge and common knowledge in a distributed environment (revised version). Technical Report RJ 4421, IBM.

Huhns, Michael N.; Bridgeland, David; and Arni, Natraj 1990. A DAI communication aide. Technical Report ACT-RA-317-90, Microelectronics and Computer Technology Corporation, AI Laboratory, Austin, TX.

McCarthy, John 1979. Ascribing mental qualities to machines. In Ringle, Martin, editor, *Philosophical Perspectives in Artificial Intelligence.* Harvester Press. Page nos. from a revised version, issued as a report in 1987.

Perrault, Raymond 1987. An application of default logic to speech act theory. Technical Report 90, Center for the Study of Language and Information, Stanford, CA.

Searle, John R. 1969. *Speech Acts.* Cambridge University Press, Cambridge, UK.

Singh, Munindar P. 1990. Group intentions. In *10th Workshop on Distributed Artificial Intelligence.*

Singh, Munindar P. 1991a. Group ability and structure. In Demazeau, Y. and Müller, J.-P., editors, *Decentralized Artificial Intelligence, Volume 2.* Elsevier Science Publishers B.V. / North-Holland, Amsterdam, Holland.

Singh, Munindar P. 1991b. Intentions for multiagent systems. Submitted; extends [Singh, 1990].

Singh, Munindar P. 1991c. A logic of situated know-how. In *National Conference on Artificial Intelligence (AAAI).*

Singh, Munindar P. 1991d. Towards a formal theory of communication for multiagent systems. In *International Joint Conference on Artificial Intelligence (IJCAI).*

Thomas, Becky; Shoham, Yoav; and Schwartz, Anton 1990. Modalities in agent-oriented programming. Computer Science Department, Stanford University.