Chapter 6

Communications

Communication among agents in multiagent systems may be fruitfully studied from the point of view of speech act theory. In order for multiagent systems to be formally and rigorously designed and analyzed, a semantics of speech acts that gives their objective model-theoretic conditions of satisfaction is needed. However, most research into multiagent systems that deals with communication provides only informal descriptions of the different message types used. And this problem is not addressed at all by traditional speech act theory or by research into discourse understanding. I provide a formal semantics for the major kinds of speech acts using the definitions of intentions and know-how that were developed in previous chapters. This connection to other theories is reason to be reassured that this theory is not *ad hoc*, and will coherently fit in a bigger picture. The resulting theory applies uniformly to a wide range of multiagent systems. Some applications of this theory are outlined and it is used to analyze the contract net protocol.

6.1 Protocols Among Agents

The behavior of a multiagent system depends not just on its component agents, but also on how they interact. Therefore, in a multiagent 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. I take *protocols* to be the specifications of these interactions. Protocols, when seen in this way, are a nice way of enforcing modularity in the design of a multiagent 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.

A formal theory of the kinds of communication that may take place among agents is crucial to the design and analysis of complex multiagent systems. Unfortunately, no theory is currently available that provides the *objective semantics* of the messages exchanged. I propose to develop such a theory by building on work in speech act theory. Before getting to the technical details, I briefly describe what speech act theory is and how it may be applied to multiagent systems.

6.1.1 Speech Act Theory

Speech Act Theory deals primarily 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 themselves. 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 & Harnish, 1979; Searle, 1969]. A speech act is associated with at least three distinct actions:

1. a *locution*, or the corresponding physical utterance,
2. an *illocution*, or the conveying of the speaker's intent to the hearer, and
3. any number of *perlocutions*, or 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, since some of them may occur in an inappropriate state of the world, e.g., when no receiver is available. At the same time, all perlocutions are not caused by appropriate illocutions, since 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]. I adopt this practice in this chapter.

A speech act is usually seen to have two parts: an *illocutionary force* and a *proposition* [Searle, 1969]. The illocutionary force distinguishes, e.g., a command from a promise; the proposition describes the state of the world that is, respectively, commanded or promised. The propositional part of an

Force	Example
Assertive	The door is shut
Directive	Shut the door
Commissive	I will shut the door
Permissive	You may shut the door
Prohibitive	You may not shut the door
Declarative	I name this door the Golden Gate

Table 6.1: Classification of Speech Acts

illocution specifies the state of the world that it is, in some sense, about. For example, an assertive asserts of that state that it holds currently (though the proposition could be temporal); a directive asks the hearer to bring that state about; a commissive commits the speaker to bringing it about, and so on. Paradigmatic examples of speech acts of different illocutionary forces are given in Table 6.1. The satisfaction of a speech act depends both on its illocutionary force and on its proposition.

The classification of speech acts given above is necessarily coarse. Speech acts of varying strengths, and of differing pragmatic effects are lumped together here. For example, assertives include statements, tellings, claims, and so on; and, directives include commands, entreaties, requests, advice, and so on. This should not be taken to mean that the proposed theory cannot accommodate different kinds of speech acts, or that it cannot capture the distinctions between, e.g., requests and commands. It just means that the distinctions between them are not seen to be *semantic.* The conditions of satisfaction of different speech acts in the same class are identical. Their differences lie in pragmatic factors, e.g., relative social stature of the agents involved and matters of cultural convention. For example, a command can be successfully issued only to subordinates; however, one can request almost anyone. Further constraints on when requests and commands are issued and satisfied may be stated that capture their non-semantic aspects properly.

6.1.2 Speech Act Theory in Multiagent Systems

There are two kinds of applications of Speech Act Theory in multiagent systems. 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 [Chang, 1991; Thomas *et al.*, 1990]. This is quite standard, and something I shall do myself. However, the traditional proposals are informal: even when they are part of a formal theory, e.g., [Thomas *et al.*, 1990], they rely on one's understanding of the labels used to assign meanings to the different message types. The true meanings are embedded in the procedures that manipulate messages of different types.

The second kind of application of Speech Act Theory involves approaches, which treat illocutions as linguistic actions and aim to describe the interactions of agents in terms of what they say to each other. They attempt to generalize linguistic theories designed for human communication to artificial systems [Cohen & Levesque, 1988b]. These theories suffer from being based on traditional formalizations of speech acts [Allen & Perrault, 1980]. Such 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 [Grice, 1969; Searle, 1975]. 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 no use in an artificial system other than for interaction with humans: multiagent 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.

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 the interesting part of the semantics of speech acts, as they may be applied in multiagent systems, concerns what they cause to be done rather than whether they are interpreted to be of one kind or another.

6.1.3 The Need for a Semantics

The formalization undertaken here concerns the objective conditions of satisfaction for different kinds of messages. Not only is this useful from the point of view of design, it also helps clarify our intuitions about the process of deliberation, since ideally the agents should act so as to "satisfy" some appropriate subset of the messages communicated in their system. The main original contributions of this chapter are described by the postulates given below.

- There is a level of formal semantics of speech acts that is distinct from both (a) what is traditionally considered their semantics, namely, the conditions under which they may be said to have occurred, and (b) their pragmatics, namely, the effects they may or ought to have on the speaker's and hearer's cognitive states. That is, the proposed semantics differs from both the illocutionary and the perlocutionary aspects of speech acts.

- The semantics of speech acts roughly corresponds to the conditions under which we would affirm that the given speech act had been satisfied.
- This semantics can be captured in the usual model-theoretic framework by introducing an operator that distinguishes the satisfaction of a speech act from its mere occurrence.
- The definitions be given in terms of the intentions and know-how of the participants and the state of the world (at some salient time or times).

The conditions of satisfaction for most kinds of speech acts differ significantly from those of assertives that are ordinarily considered in logic. Assertives, being claims of fact, are true or false; other speech acts call for a more complex notion of success. In the context of imperatives, Hamblin distinguishes between what he calls *extensional* and *whole-hearted* satisfaction [Hamblin, 1987, pp. 153–157]. Briefly, the former notion admits accidental success, while the latter does not. Hamblin's aim was simply to be able to state prescriptive conditions on when what kind of imperatives ought to be issued, and the philosophical problems that arise when one is in a "quandary." That is, his focus was pragmatic. I take advantage of some of his ideas, but make a finer distinction and extend it to other important kinds of speech acts here, formally relating them to intentions and know-how in the process.

In section 6.2, I formalize the notion of satisfaction that I argue is appropriate for multiagent systems. In section 6.3, I show how this framework may be used in the design of multiagent systems by using it to state constraints on communication and to formalize the contract net protocol.

6.2 Formal Model and Language

6.2.1 Speech Acts as Actions

Speech acts are, first of all, actions. I take them to be the actions of their speakers, and as occurring over periods (the same as actions in general). The reader may think of the receiver as listening over a part of the period during which the sender is speaking. This is not used in the formalization, however.

Let **says-to** be a parametrized speech act, to be used as in **says-to**(y, m). $[S; t_b, t_e] \in [\![\textbf{says-to}(y, m)]\!]^x$ means that, on scenario S, agent x performed the speech act of saying m to agent y in the time from t_b (the moment of beginning) to t_e (the moment of ending). This means that the illocution was successfully made. There is no commitment at this stage as to whether it was satisfied or not. Recall that $[S; t, t']$ presupposes that $t \leq t'$.

The semantics of speech acts is captured in the theory of this chapter by means of a modal operator, W. It is convenient to have a special predicate in the language that allows us to talk of the performance of a speech act. This allows us to apply the modal operators to formulae that denote propositions, rather than to those that denote actions. Besides allowing us to follow the usual way of defining a modal operator, the definition of **comm** also allows speech acts to be nested as in "I tell you that he pleaded guilty."

Let the new predicate be **comm** that applies to two agents, and an illocution. Since actions take place over scenarios, it is most convenient to evaluate **comm**(x, y, m) at scenarios and moments. **comm**(x, y, m) is true at S, t just if y said (or started to say) m to x then. A performed illocution may, of course, not be satisfiable. For example, some commands may be issued that are impossible to obey. The operator W, then, applies on formulae of the form **comm**(x, y, m). It denotes the whole-hearted satisfaction of the given speech act.

Whole-Hearted satisfaction is defined relative to a scenario and a moment. A performative is taken to be in force as soon as it is completed (and not sooner). This is done to allow the possibility of a communication being aborted midway. That is, a speaker's failed attempts to say something, i.e., to get his point across, do not count as communications.

6.2.2 Formal Language

The formal language of this chapter, $\mathcal{L}^m$ is $\mathcal{L}^c$ augmented with the operator, W. In the following, $\mathcal{F}$ = {assertive, directive, commissive, permissive, prohibitive, declarative} is the set of illocutionary forces. $\mathcal{M}$ is the set of *messages* as defined below. The set of basic actions, $\mathcal{B}$, is extended with illocutionary actions, which are generated from the messages and are formally treated as the actions of the sending agent. The resulting set is called $\mathcal{B}^m$.

SYN-31. All the rules for $\mathcal{L}^c$ with $\mathcal{L}^m$ substituted for $\mathcal{L}^c$ and $\mathcal{B}^m$ substituted for $\mathcal{B}$

SYN-32. All the rules for $\mathcal{L}_s^c$ with $\mathcal{L}_s^m$ substituted for substituted for $\mathcal{L}_s^c$ and $\mathcal{B}^m$ substituted for $\mathcal{B}$

SYN-33. All the rules for $\mathcal{L}_y^c$ with $\mathcal{L}_y^m$ substituted for substituted for $\mathcal{L}_y^c$

SYN-34. $p \in \mathcal{L}_s^m$ implies that $\mathsf{W}p \in \mathcal{L}_s^m$

SYN-35. $p \in \mathcal{L}^m$ and $i \in \mathcal{F}$ implies that $\langle i, p\rangle \in \mathcal{M}$

SYN-36. $x, y \in \mathcal{A}$ and $m \in \mathcal{M}$ implies that $\mathbf{comm}(x, y, m) \in \mathcal{L}_s^m$

SYN-37. $x, y \in \mathcal{A}$ and $m \in \mathcal{M}$ implies that $\textbf{says-to}(x, y, m) \in \mathcal{B}^m$

6.2.3 Whole-Hearted Satisfaction

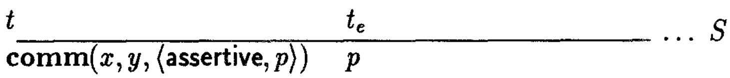

Figure 6.1: The Satisfaction Condition for Assertives

SEM-53. **Assertives:**

$M \models_{S,t} \mathsf{W}(\mathbf{comm}(x, y, \langle\mathsf{assertive}, p\rangle))$ iff
$(\exists t_e : [S; t, t_e] \in [\![\textbf{says-to}(x, y, \langle\mathsf{assertive}, p\rangle)]\!]^x$ and $M \models_{S,t_e} p)$

An assertive is satisfied simply if its proposition is true at the moment the utterance is made. Thus the assertive, "The door is shut," is satisfied on all scenarios where the door is, in fact, shut. The satisfaction conditions for the other kinds of speech acts are more interesting than this.

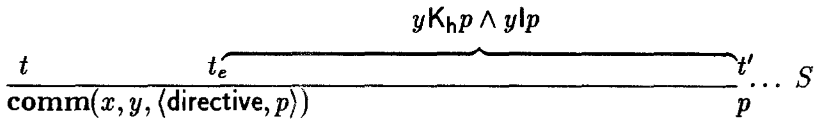

Figure 6.2: The Satisfaction Condition for Directives

SEM-54. **Directives:**

$M \models_{S,t} \mathsf{W}(\mathbf{comm}(x, y, \langle \mathsf{directive}, p \rangle))$ iff
$(\exists t_e : [S; t, t_e] \in [\![\mathbf{says\text{-}to}(x, y, \langle \mathsf{directive}, p \rangle)]\!]^x$ and $M \models_{S,t_e} (y\mathsf{K_h}p \wedge y\mathsf{I}p)\mathsf{U}p)$

A directive is satisfied just if (a) its proposition, p, becomes true at a moment in the future of its being said, and (b) all along the scenario from now to then, the hearer has the know-how, as well as the intention to achieve it. For example, a directive to open the door is satisfied if the door ends up open (within some salient period of time, perhaps), and furthermore the hearer continuously planned to open the door and was in a position to be able to execute the plan to open it. Note that this definition does not finally require that the door open because of the hearer's actions. This would not be an important requirement to impose in my view, and would only cause action-theoretic complications about the matter of when an agent can be said to have performed a certain action, especially when that action is not a single-step basic action.

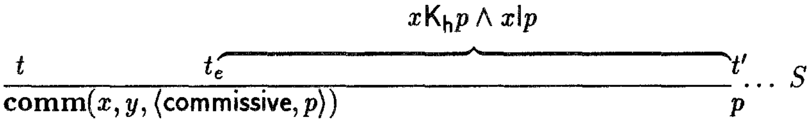

Figure 6.3: The Satisfaction Condition for Commissives

SEM-55. **Commissives:**

$M \models_{S,t} \mathsf{W}(\mathbf{comm}(x, y, \langle \mathsf{commissive}, p \rangle))$ iff
$(\exists t_e : [S; t, t_e] \in [\![\mathbf{says\text{-}to}(x, y, \langle \mathsf{commissive}, p \rangle)]\!]^x$ and $M \models_{S,t_e} (x\mathsf{K_h}p \wedge x\mathsf{I}p)\mathsf{U}p)$

Similarly, a commissive is satisfied just if (a) its proposition becomes true at a moment in the future of its being said, and (b) all along the scenario from now to then, the *speaker* has the know-how, as well as the intention to achieve it. Technically, a commissive is just like a directive except that the role of the hearer is taken over by the speaker. For example, the commissive, "I promise to shut the door," is satisfied on all scenarios on which the door eventually gets shut and until it does, the speaker intends and knows how to shut it.

A difference with directives that is of significance in some applications is that the satisfaction condition of a commissive depends on the actions, intentions, and know-how of just one agent. This can make the satisfaction of commissives easier to enforce in artificial systems. A related observation that is also interesting is that there seem to be fewer forms of commissives in natural languages than directives. This seems to be related to the fact that the satisfaction of directives involves actions by agents other than the speaker, and so different kinds of social considerations come into play. One may request or command or beseech or advise someone to do something, but one can just do it on one's own (though threats can express commitments conditional on the hearer's actions).

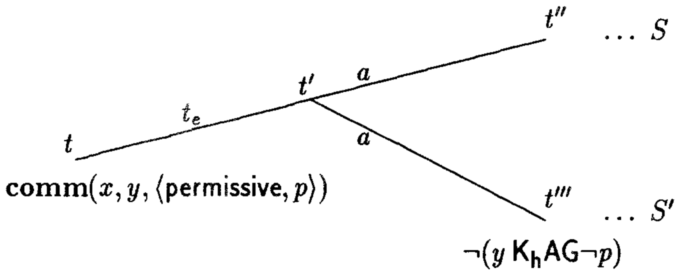

Figure 6.4: The Satisfaction Condition for Permissives

SEM-56. **Permissives:**

$M \models_{S,t} \mathsf{W}(\mathbf{comm}(x, y, \langle \mathsf{permissive}, p\rangle))$ iff
$(\exists t_e : [S; t, t_e] \in [\![\mathbf{says\text{-}to}(x, y, \langle \mathsf{permissive}, p\rangle)]\!]^x$ and $(\exists t' : t_e \leq t'$ and
$(\forall a : M \models_{S,t'} y\langle a\rangle\mathsf{true} \Rightarrow M \models_{t'} \mathsf{E}y\langle a\rangle\neg y\mathsf{K_h}(\mathsf{AG}\neg p))))$

A permissive is satisfied at a scenario and a moment just if it is taken advantage of by the hearer at a future moment on that scenario. But when a permissive is taken advantage of, it allows the hearer to do actions at certain times that he could not have done before, because they might possibly have led to the condition becoming true. Thus a permissive is satisfied on a scenario on which the hearer does at least one action whose performance can lead to a state where he is unable to prevent that condition from occurring. That is, the hearer can now risk letting that condition hold. For example, a permissive allowing a hearer to let the door be open is satisfied on a scenario, if (as a result of the given permissive, as it were), the hearer can, e.g., risk opening the window, even though the breeze may open the door. Without this permissive, the hearer would have to take some precaution, e.g., latch the door, before opening the window. The satisfaction of a permissive tends to increase the know-how of the hearer by giving him more options. Unfortunately, no closed-form characterization of this increase in know-how is available at present.

The notion of prevention is captured here using the previous definition of know-how: an agent, x, can prevent p iff he knows how to achieve $\mathsf{AG}\neg p$. This is perhaps too strong in that it requires the agent to arrive at a state where p becomes impossible on every scenario. An alternative definition would let p be possible on scenario, but ensure that is always avoidable through some action on part of the agent. However, the present definition has the advantage of permitting only a finite number of actions. I expect some variation in this component of the semantics depending on what notion of prevention is actually plugged in. The present version appears good enough, but may eventually need to be refined to use the more general notion of prevention where the agent continually prevents the given condition.

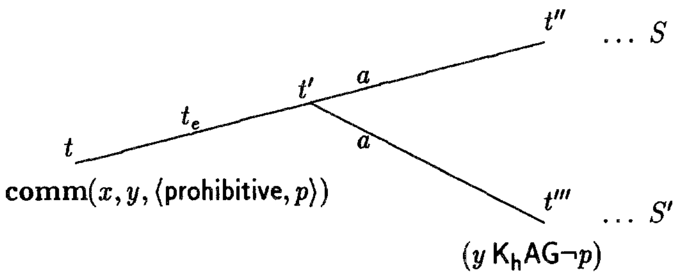

Figure 6.5: The Satisfaction Condition for Prohibitives

SEM-57. **Prohibitives:**

$M \models_{S,t} \mathsf{W}(\mathbf{comm}(x, y, \langle \mathsf{prohibitive}, p\rangle))$ iff
$(\exists t_e : [S; t, t_e] \in [\![\mathbf{says\text{-}to}(x, y, \langle \mathsf{prohibitive}, p\rangle)]\!]^x$ and $(\forall t_e \leq t' : (\forall a :$
$M \models_{S,t'} y\langle a\rangle\mathsf{true} \Rightarrow M \models_{t'} \mathsf{A}y\langle a\rangle y\mathsf{K_h}(\mathsf{AG}\neg p))))$

A prohibitive is satisfied at a scenario and moment just if none of the actions done by the hearer on that scenario (in the future), can lead to a state where the hearer would be unable to prevent the condition from occurring. That is, the hearer cannot risk violating the prohibition. In other words, the hearer should always (on the given scenario) know how to prevent the prohibited condition and the prohibited condition should not occur on the given scenario. For example, a prohibitive to not let the door be open can be satisfied only if the hearer does not let the window be open, where the opening of the window may lead to the door being opened.

$$t\ \overbrace{\underline{\hspace{20em}}}^{(x\mathsf{K}_{\mathsf{h}}p) \wedge (x\mathsf{I}p)}\ t_e \ldots S$$
$$\mathbf{comm}(x, y, \langle \mathsf{declarative}, p \rangle) \hspace{12em} p$$

Figure 6.6: The Satisfaction Condition for Declaratives

SEM-58. **Declaratives:**

$M \models_{S,t} \mathsf{W}(\mathbf{comm}(x, y, \langle \mathsf{declarative}, p \rangle))$ iff $(\exists t_e : [S; t, t_e] \in [\![\mathbf{says\text{-}to}(x, y, \langle \mathsf{declarative}, p \rangle)]\!]^x$ and $M \models_{S,t_e} p$ and $(\forall t' : t \leq t' < t_e \Rightarrow M \not\models_{S,t'} p)$ and $M \models_{S,t} (x\mathsf{K}_{\mathsf{h}}p \wedge x\mathsf{I}p)\mathsf{U}p)$

A declarative is satisfied just if (a) its proposition, p, becomes true for the first time at the moment that it is said, and (b) all along while the speaker is saying it, he intends that condition to occur and knows how to make it occur. For example, a declarative to name a certain door the Golden Gate is satisfied if the door ends up named thus, and the speaker intended it to be so named and knew how to name it. The door has its new name as soon as the declarative is completed. The condition about the know-how is included to ensure that, at each moment, the speaker is able to force the completion of the declarative and thereby force the occurrence of the appropriate condition. This helps eliminate cases where the speaker has the intention, but is not in the right social or conventional position to make the declarative succeed. In our example, the naming should succeed, but not because of some contingent features of the given scenario. According to some traditional theories, e.g., that of Vanderveken [1990], the occurrence of declaratives coincides with their success. This seems too weak since it allows a declarative to succeed even if the speaker did not have full control over its occurrence, i.e., even if the speaker could not have forced the given condition to occur.

6.2.4 Interrogatives

The theory as developed so far must be extended to account for interrogatives (and replies). This extension is needed so that this theory may be used for applications where agents interact by querying and replying to one another. What distinguishes interrogatives from other speech acts is the fact that their satisfaction necessarily requires the hearer to perform some speech act. In particular, questions are satisfied when their (true) answer is supplied by the hearer to the speaker.

In order for answers to be coherently defined, I need to extend the definition of the proposition contained in a message to allow for structures to which answers may be specified. The obvious extension is to allow lambda expressions. For a question of this form, the answer is naturally defined as the set of objects (of some sort or from some set) for which the lambda expression evaluates to true. For lambda expressions with more than one argument, the answer is the set of tuples over which it evaluates to true. This definition of an answer is not only a natural one in cases where knowledge-bases are used, but is also compatible with research about the semantics of natural language questions [Groenendijk & Stokhof, 1984].

Thus a question is interpreted in the semantics as if it were a directive to perform an assertive speech act (back to the speaker of the question) that provides the answer to the question. Therefore, a question is satisfied when a true answer to it is given, i.e., when the assertive containing the answer is itself satisfied. This derives from an intuition about treating interrogatives as imperatives of the form 'Tell me Truly' [Harrah, 1984, pp. 747–748]. Let a message be a pair, $\langle \mathsf{interrogative}, (\lambda\vec{a}p)\rangle$; since answers are assertives, we just need a new predicate $\mathbf{answer}((\lambda\vec{a}p), \mathbf{Ans})$, where $|\mathbf{Ans}|$ is finite and $(\forall\vec{b} : (\vec{b} \in \mathbf{Ans}) \leftrightarrow (p|^{\vec{a}}_{\vec{b}}))$, i.e., $\mathbf{Ans}$ is the answer. The know-how to produce an answer that is required of the hearer would involve not just the physical capabilities needed to make an assertive utterance, but also the knowledge needed to compute the answer. Let r abbreviate $\mathsf{W}(\mathbf{comm}(y, x, \langle \mathsf{assertive}, \mathsf{answer}((\lambda\vec{a}p), \mathbf{Ans})\rangle))$.

In order to capture this in the formal language, we need to introduce a set of constants, $\mathcal{D}$, and a set of predicates, $\mathcal{PRED}$. $\mathcal{L}^q_q$ denotes the set of query expressions. We have the following syntax, where $\mathcal{L}^q$ is the version of the formal language that includes interrogatives.

SYN-38. All the rules for $\mathcal{L}^m$ with $\mathcal{L}^q$ substituted for $\mathcal{L}^m$

SYN-39. All the rules for $\mathcal{L}^m_s$ with $\mathcal{L}^q_s$ substituted for $\mathcal{L}^q_s$

SYN-40. All the rules for $\mathcal{L}^m_y$ with $\mathcal{L}^q_y$ substituted for $\mathcal{L}^m_y$

SYN-41. $\vec{b} \in (\mathcal{D} \cup \mathcal{X})^n$ and $P \in \mathcal{PRED}$ implies that $P(\vec{b}) \in \mathcal{L}^q$

SYN-42. $u \in \mathcal{X}$ and $p \in \mathcal{L}^q$ implies that $(\exists u : p) \in \mathcal{L}^q$

SYN-43. $\vec{a} \in \mathcal{X}^n$, $p \in \mathcal{L}^q$ implies that $(\lambda\vec{a} : p) \in \mathcal{L}^q_q$

SYN-44. $p \in \mathcal{L}^q_q$ implies that $\langle \mathsf{interrogative}, p\rangle \in \mathcal{M}$

SYN-45. $\langle \mathsf{interrogative}, p\rangle \in \mathcal{M}$ and $\mathbf{Ans} \in \mathcal{D}^n$ implies that $\mathbf{answer}(p, \mathbf{Ans}) \in \mathcal{L}^q$

SEM-59. $M \models_{S,t} \mathsf{W}(\mathbf{comm}(x, y, \langle \mathsf{interrogative}, (\lambda\vec{a}p)\rangle))$ iff
$(\exists t_e : [S; t, t_e] \in [\![\mathbf{says\text{-}to}(x, y, \langle \mathsf{interrogative}, (\lambda\vec{a}p)\rangle)]\!]^x$ and
$M \models_{S,t_e} (y\mathsf{K_h}r \wedge y\mathsf{I}r)\mathsf{U}r)$

This takes care of the so-called Wh-questions; Yes-no questions are analogous. A yes-no question for q is a directive to truthfully assert q or $\neg q$. The details are not included here. In some applications, it is useful to allow questions whose answers are commissives (e.g., when the question is a call for a bid section 6.3) or directives (e.g., when the question is a call for advice). The above definitions can also be extended to allow for questions whose answers are given in a piecemeal manner; e.g., as by Prolog interpreters. This would also allow the answers to be evaluated lazily.

6.3 Applying the Theory

The two main motivations for developing the above theory were to provide a rigorous foundation for the design of multiagent systems and to justify some prescriptive claims about how agents should communicate in such systems. The proposed definitions give objective criteria with which to evaluate the correctness of the different scenarios that are the possible runs or executions of a multiagent system. In design, the problem is to create a system which allows only correct scenarios to be realized. Prescriptive claims for agents tell them what to do given their beliefs and intentions, so that only correct scenarios may emerge.

The definition of W can be used to motivate some correctness conditions for multiagent systems. A scenario may be defined to be correct if all the messages passed on it are satisfied. In general terms, the designer's goal is to ensure that all runs that may be realized are correct. This reduces to the design goal that the intentions and know-how of the agents be such that only correct scenarios are realized. This is the sense of correctness that designers use in practice. They usually achieve this kind of correctness by a number of means, e.g., hard-wiring the intention to cooperate in their agents, or by setting up appropriate hierarchical structures. These structures may ensure different patterns of interaction, e.g., that some directives (commands) are always obeyed, and others (requests) obeyed whenever they do not conflict with the hearer's current intentions.

6.3.1 Normative Constraints on Communication

The ways in which a semantics of speech acts, such as the one developed here, may be applied in multiagent systems are perhaps obvious. A semantics can lead to a clearer understanding of the issues involved in the functioning of multiagent 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 constrain his designs so that they allow only correct scenarios to be followed. 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.

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 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 formulae (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, 1990, pp. 1058–

1063] and [Burch *et al.*, 1990]. For the particular logic developed here, such automated methods are not yet available.

It should be clarified that the propositions used in the messages are descriptions of conditions of the world, or of the agents' 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 this 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 be quite reasonable from Bill's point of view. The propositions are evaluated at moments in the model, and may have different truth values at different such moments. The time of reference (e.g., "6:00 pm") could be specified as part of 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. Of course, some of those objective conditions could be about the beliefs of agent; that is, both p and $x\mathsf{B}p$ may be specified, but they are distinct propositions. 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 moments. This too is something that the designer must ensure, if the designed system is to function as desired.

I now give some examples of formalizations of design constraints. It is by no means suggested that all these constraints make sense in all applications: they are stated below only to exhibit the power of the theory. In the next section, I discuss an extended example that shows how constraints such as these may be used in multiagent systems.

COMM-1. **Intending One's 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.

$$\mathbf{comm}(x, y, \langle \mathsf{directive}, p\rangle) \rightarrow x\mathsf{I}p$$

COMM-2. **Preference for Local Action:**

If an agent knows how to achieve a proposition by himself, he should not issue it as a directive. For example, an agent who needs to raise the voltage on a part of a power network he jointly controls with another agent should do so by himself, rather than request the other agent to do so. This constraint is especially useful when communication is expensive or introduces substantial delays.

$x\mathsf{K_h}p \rightarrow \neg\mathbf{comm}(x, y, \langle\mathsf{directive}, p\rangle)$

COMM-3. **Load-Dependent Preference for Local Action:**

In practice, constraint COMM-2 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 he carries out the actions that he has to do to fulfill other commitments. Thus an agent may request another agent to do a task that he would have done himself, had he not been swamped with other tasks.

$x * Y \wedge x[\langle Y\rangle]p \Rightarrow \neg\mathbf{comm}(x, y, \langle\mathsf{directive}, p\rangle)$

In other words, if an agent has a strategy by following which he knows how to achieve p, then he does not request another agent to achieve p. This is because, unless he gives up his strategy, he will in fact succeed in achieving p. This is a consequence of Theorem 5.6.

COMM-4. **Weak Consistency for Directives:**

A directive issued by an agent should not clash with the agent's own intentions. That is, a speaker's intentions and his directives should be compatible, at least in some scenarios. For example, if an agent intends that the voltage V_1 decrease, then he should not even request another agent to raise voltage V_2, if raising voltage V_2 would necessarily raise V_1 as well. This constraint differs significantly from constraint COMM-1. Constraint COMM-1 says that the issuer intends the given directive; this constraint says that all of the issuer's intentions are consistent with the directive.

$x\mathsf{I}q \wedge \mathbf{comm}(x, y, \langle\mathsf{directive}, p\rangle) \rightarrow$
$\mathsf{E}(\mathsf{W}\mathbf{comm}(x, y, \langle\mathsf{directive}, p\rangle) \wedge \mathsf{F}q)$

COMM-5. **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 his intentions. That is, on all scenarios on which the directive is satisfied, the speaker should eventually know how to achieve his intentions. In fact, the formalization given below allows 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 either be 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.

$$x\mathsf{I}q \wedge \mathbf{comm}(x, y, \langle \mathsf{directive}, p\rangle) \rightarrow$$
$$\mathsf{A}(\mathsf{W}\mathbf{comm}(x, y, \langle \mathsf{directive}, p\rangle) \rightarrow \mathsf{F}x\mathsf{K_h}q)$$

COMM-6. **Weak Consistency for Prohibitives:**

A prohibitive is issued by an agent only if the agent himself 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 at that moment intend that the latter connect to it. Recall the discussion on propositions earlier in this section. Thus the agent who prohibited the other from connecting to an outlet might himself intend to connect to that outlet; however, there is no problem here, since the two propositions are distinct.

$$\mathbf{comm}(x, y, \langle \mathsf{prohibitive}, p\rangle) \rightarrow \neg x\mathsf{I}p$$

COMM-7. **Weak Consistency for Permissives:**

A permissive is issued by an agent only if the agent himself does not intend that the relevant proposition never occur. That is, the agent who permits another from letting a certain condition occur should not himself 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 he should not permit others to let it hold. For example, if an agent intends to keep a certain power outlet available for his own use, he should not permit others to use it, because that could only render it unavailable at certain times in the future.

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

Certain examples that may seem to contradict the applicability of this constraint actually do not: they just have to be formalized carefully. One case involves game playing, where an agent seemingly permits another to beat him, but intends to win nonetheless. While the above constraint is meant only as an example and need not apply in all cases, in this particular case, the permissive is simply 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 being realized. Once a game begins, the two agents are peers and neither can permit or prohibit actions of the other.

COMM-8. **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 should never be 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, he should not require another agent to prevent that very condition. The latter's success essentially precludes the former from succeeding with the directive.

$$\neg \mathsf{E}(\mathsf{Fcomm}(x, y, \langle \mathsf{directive}, p \rangle) \wedge \mathsf{Fcomm}(x, z, \langle \mathsf{prohibitive}, p \rangle))$$

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 conditions to be risked: they do not require them to occur.

COMM-9. **Prior Commitment:**

A directive should be issued only after a conditional promise is given by the intended receiver that he 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 application, 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.

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

6.3.2 The Contract Net

The Contract Net Protocol of Davis & Smith is among the most well-known and significant protocols for multiagent systems [Davis & Smith, 1983]. While several variations of it are possible, in its most basic form it may be described as in Figure 6.7. We are given a system with several agents. One of them has a task that he has to perform. He 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*. He 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. He 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, he communicates the *result* of performing the assigned task to the manager. The manager collects the results of all the subtasks of the original task and thus computes its result. If that task was assigned to him by some other agent, he then sends the result to that agent.

The key steps in the contract net protocol from our point of view are the following: (a) the call for bids, (b) the bids, (c) the assignment of the task, and (d) 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. I 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."

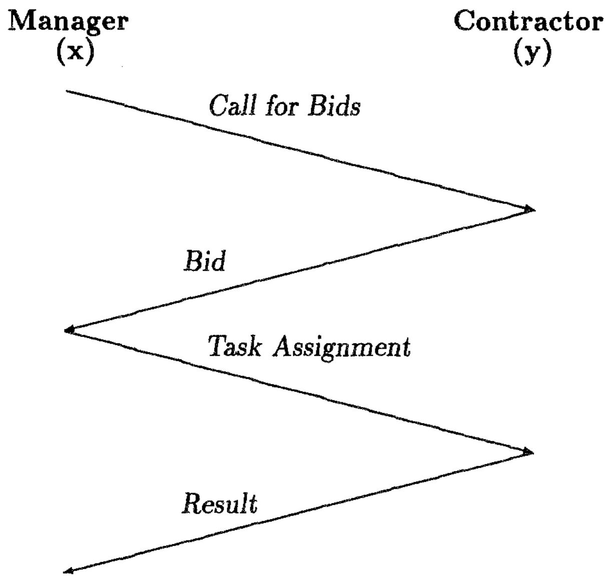

Figure 6.7: Messages Exchanged in the Contract Net

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 **comm**$(y, x, \langle$**assertive**, result$(T)\rangle)$ (*result*); let $a(x, y, T)$ abbreviate **comm**$(x, y, \langle$**directive**, $r(x, y, T)\rangle)$ (*assignment*); and let $b(x, y, T)$ abbreviate **comm**$(y, x, \langle$**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 of them are satisfied vacuously, if $\chi(x, y, T)$ is false.

- $\langle$**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 him to send him 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 he can perform the given task. As an aside, note that a contractor should not bid on two or more tasks he 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, provided some additional assumptions are made. Below, 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 realized 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.

6.4 Conclusions

I now compare the approach discussed above to some semantics of speech acts that others have proposed. One important work is that of Searle & Vanderveken [1985]. However, they do not relate the satisfaction conditions of different sorts of speech acts with the intentions and know-how of the speaker or the hearer. Their greater aim seems to be to derive the possible illocutionary forces from a set of core features, e.g., what they call *illocutionary point* and *direction of fit.*

Searle & Vanderveken's approach has been challenged by Cohen & Levesque who argue that the illocutionary point is theoretically redundant and can be derived from the inferences that a speech act sanctions [Cohen & Levesque, 1988b]. These inferences involve the updating of the beliefs, intentions, and mutual beliefs of the speaker and the hearer. For this reason, Cohen & Levesque's approach is largely of pragmatic interest. Perrault has argued that, despite Cohen & Levesque's attempts, how the participants' cognitive states ought to be updated cannot be monotonically specified [Perrault, 1987]. He proposes that a default mechanism, in his paper Reiter's default logic, be used to characterize the effects of speech acts and, hence, their pragmatic content. The effects of speech acts are related to the processes of deliberation 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-revision and intention-revision, which are still not sufficiently well-developed. In general, these processes depend on issues like the social relationship of the agents or on matters of performance, rather than on the semantics of communication *per se.* Perrault suggests some postulates for such revision using default logic. Thus his focus is not on the semantics as considered here.

In any case, a semantics would help clarify our intuitions even about the pragmatic aspects of communication. As a clarification of my goals, note that the role of the proposed semantics is akin to that of classical semantics for assertives. Classical semantics only tells us when an assertive is objectively satisfied: it makes no claims about when an assertive should actually be uttered or believed.

Werner has proposed a theory of communication based on this theory of intentions, which was discussed in section 3.5 also [Werner, 1989]. Werner considers only directives (besides assertives), and defines their effects on the hearer's "intentional state." A directive forces the hearer's intentional state so that it would be necessarily satisfied no matter what the hearer does (according to the modified intentional state). Werner thus seems to be attacking the problem of the effects of directives in idealized circumstances. A notable weakness of this theory is the lack of compositionality: operators like $\wedge, \vee$, and $\neg$ mean differently in the context of directives than otherwise.

In more recent work than his book with Searle, Vanderveken has independently addressed the problems of the "success and satisfaction" of speech acts [1990; 1991]. Vanderveken's goal is a general illocutionary logic, and a large part of his theory is focused on the conditions of when a performative succeeds, i.e., when a speech act of a particular illocutionary force is made. His goal is to give the semantics of performative verbs in an extension of Montague grammar. He also considers the degree of strength of different speech acts explicitly, and classifies a variety of speech act verbs, as special as the declaratives, "homologate" and "ratify," which differ primarily on their pragmatic aspects. The particular definitions given by Vanderveken are extensional in that no reference is made to the intentions or the know-how of the agents. For example, for him a directive is satisfied if the appropriate condition comes to hold, and a prohibitive, merely a special kind of directive for him, is satisfied if the appropriate condition does not occur. He lumps permissives and prohibitives with directives [Vanderveken, 1990, pp. 189–198], which I have argued should not be done. Vanderveken also does not consider the temporal aspect of speech acts explicitly. In sum, while the results of the theory developed here are more refined than the corresponding results of his theory, they could fruitfully be combined with the pragmatic and other aspects of speech acts that he has studied in much greater detail.

The proposed taxonomy of speech acts is motivated by the semantic definitions given above, which are different for permissives, prohibitives, and directives. This distinguishes the proposed taxonomy from other classifications of speech acts. Since syntactically, permissives, prohibitives, and directives are all imperatives, they are usually classified together, e.g., by Bach & Harnish [1979, pp. 39–54] and Searle & Vanderveken [1985, ch. 9]. This is surprising in the case of Searle & Vanderveken, since their interests are pragmatic, rather than syntactic.

The relationship between the proposed approach and traditional work on speech acts in natural language processing (NLP) is essentially one of complementarity. Traditional theories address the problem of determining when what kind of a speech act occurs. They can thus be used to feed into the proposed theory. One simply has to use the NLP theories under appropriate assumptions to determine the truth of different instances of $\mathbf{comm}(x, y, m)$ and then apply the proposed theory to determine the satisfaction conditions of those expressions. This perspective places the semantics presented here at the natural boundary of deciding what to say, on the one hand, and deciding how to say it, on the other. That is, on the one hand, we have the concerns of deciding what speech act to make, and on the other, the concerns of how to get a point across. This is a useful way to organize a multiagent system that is designed to also communicate with humans: the first aspect mentioned above is a part of distributed computing, the second aspect a part of natural language processing.

An attractive feature of this approach is that it brings the satisfaction conditions for speech acts into the fold of logic. Using definitions of the intentions and know-how of an agent, I was able to give rigorous definitions of the conditions of satisfaction for speech acts of different of illocutionary forces. The theory presented here can yield some normative constraints on communication among agents. An advantage of the model-theoretic approach is that it allows our intuitions to be expressed directly and formally and thus can be used in clarifying and debugging them.