PLANNING NATURAL-LANGUAGE UTTERANCES TO SATISFY MULTIPLE GOALS

Technical Note 259

March 1982

By: Douglas E. Appelt, Computer Scientist

Artificial Intelligence Center
Computer Science and Technology Division

This report is a slightly revised version of a thesis submitted to the Department of Computer Science of Stanford University on December 10, 1981, in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Approved:

Nils J. Nilsson, Director
Artificial Intelligence Center

David H. Brandin, Vice President and Director
Computer Science and Technology Division

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **MAR 1982**	2. REPORT TYPE	3. DATES COVERED **00-03-1982 to 00-03-1982**

4. TITLE AND SUBTITLE **Planning Natural-Language Utterances to Satisfy Multiple Goals**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **SRI International,333 Ravenswood Avenue,Menlo Park,CA,94025**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S) 11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**190**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

ABSTRACT

PLANNING NATURAL-LANGUAGE UTTERANCES TO SATISFY MULTIPLE GOALS

This dissertation presents the results of research on a planning formalism for a theory of natural-language generation that will support the generation of utterances that satisfy multiple goals. Previous research in the area of computer generation of natural-language utterances has concentrated two aspects of language production: (1) the process of producing surface syntactic forms from an underlying representation, and (2) the planning of illocutionary acts to satisfy the speaker's goals. This work concentrates on the interaction between these two aspects of language generation and considers the overall problem to be one of refining the specification of an illocutionary act into a surface syntactic form, emphasizing the problems of achieving multiple goals in a single utterance.

Planning utterances requires an ability to reason in detail about what the hearer knows and wants. A formalism, based on a possible-worlds semantics of an intensional logic of knowledge and action, was used for representing the effects of illocutionary acts and the speaker's beliefs about the hearer's knowledge of the world. Techniques are described that enable a planning system to use the representation effectively.

The language-planning theory and knowledge representation are embodied in a computer system called KAMP (**K**nowledge **A**nd **M**odalities **P**lanner), which plans both physical and linguistic actions, given a high-level description of the speaker's goals.

The research has application to the design of gracefully interacting computer systems, multiple-agent planning systems, and the planning of knowledge acquisition.

This report is dedicated to John Gaschnig,

A brilliant scientist,
An energetic, hard-working colleague,
A good friend,

whom I have had the pleasure of knowing and working with, and who has contributed greatly, both professionally and personally, to the SRI International Artificial Intelligence Center.

ACKNOWLEDGEMENTS

It is with great trepidation that I proceed to acknowledge the contributions that others have made that culminated in the production of this thesis because so many people have contributed to the development of the ideas, and to my general well-being, that I fear I may insult someone by leaving him or her out. If this is the case, I apologize.

This work was financially supported by the Office of Naval Research under contract N00014-80-C-0296. Of course, the views and conclusions presented herein are my own and are not necessarily representative of the Office of Naval Research, or the U.S. Government. I express my gratitude to the Xerox Palo Alto Research Center for providing me with the computing resources on which the system described in this thesis was implemented. Without the contribution of Xerox's facilities, the implementation would never have been completed. I give special thanks to Dick Burton for making these facilities available to me, and Ron Kaplan, Larry Masinter, and Beau Sheil for answering my questions about and fixing bugs in INTERLISP-D.

I thank Terry Winograd for serving as my principal thesis advisor. His thoughtful comments and insights have contributed greatly to the quality of this work. I also thank Gary Hendrix, Doug Lenat, and Nils Nilsson for serving on my reading committee. All of them have provided me with discussions about and comments on earlier drafts of this thesis. Also, Barbara Grosz has given me much help and many insightful comments and has effectively served in the capacity as a reading committee member in all duties except signing the final draft. Without the help of Barbara and my committee, the final draft of this thesis would be nowhere near

as good as it is. I thank Nils in his capacity as director of the SRI International Artificial Intelligence Center for providing me with an exciting, stimulating environment in which to carry out my research. Gary has been a friend and mentor since shortly after my arrival at Stanford and is responsible for introducing me to the research opportunities at SRI International (SRI). I have had many interesting discussions with members of the natural language research group at SRI, including Armar Archbold, Barbara Grosz, Norm Haas, Gary Hendrix, Jerry Hobbs, Kurt Konolige, Bil Lewis, Paul Martin, Bob Moore, Ann Robinson, Jane Robinson, Stan Rosenschein, Daniel Sagalowicz, and Don Walker.

I acknowledge the Stanford Computer Science Department (CSD) for fostering creativity, providing financial support, and providing a truly outstanding faculty that has enhanced my understanding of all areas of computer science. I believe Stanford's Computer Science Department deserves its rating as number one in the world. I also acknowledge the contribution of someone who does much work and often receives little credit for it. Carolyn Tajnai, the Administrative Assistant for Academic Affairs, in addition to being a good personal friend, has served above and beyond the call of duty in guiding students through the bureaucratic maze and generally "running the show" in the Stanford CSD for the past three years. She has performed a very difficult job admirably over the years and has contributed greatly to my welfare and that of every student and faculty member in the department. Although she has now left to accept a long-deserved promotion, she remains one of the fondest memories of my years at Stanford.

Finally, I acknowledge the contribution of all my friends for preventing me from taking all this Piled Higher and Deeper stuff too seriously. Were it not for cross country ski trips, rock climbing, backpacking expeditions, backgammon, poker games, string quartets, and uncountable parties, this work would have been completed years ago.

TABLE OF CONTENTS

I

INTRODUCTION

0. Building the Ultimate Language-Generation System

A primary goal of natural-language-generation research in artificial intelligence is to design a system that is capable of producing utterances with the same fluency as that of a human speaker. One could imagine a "Turing Test" of sorts where a person was presented with a dialogue between a human and a computer and asked to identify which participant was the computer on the basis of the naturalness of its use of the English language. Unfortunately, no natural-language generation system yet developed can pass the test.

A language-generation system capable of passing this test would obviously have a great deal of syntactic competence. It would be capable of using correctly and appropriately such syntactic devices as conjunction and ellipsis; it would be competent at fitting its utterances into a discourse, using pronominal references where appropriate, choosing syntactic structure consistent with the changing focus, and giving an overall feeling of coherence to the discourse. The system would have a large knowledge base of basic concepts so that it could converse about any situation that arises naturally in its domain.

However, even if a language generation system met all the above criteria, it might still not be able to pass our "Turing Test" because to know only about the syntactic and semantic rules of the language is not enough. One must constantly bear in mind that language behavior is part of a coherent plan, and is directed toward satisfying the speaker's goals. Furthermore, sentences are not straightforward actions that satisfy only a single goal. When people produce utterances, their utterances are crafted with great sophistication to satisfy multiple goals at different communicative levels. For example, in a single utterance a speaker may inform a hearer of two or more propositions, make a request, shift the focus of the discourse, and flatter the hearer. On the surface, this does not argue that anything more than the above criteria is needed to produce natural-sounding utterances — all that is needed is to allow for greater complexity. Things are not that simple, however, because recognizing how an utterance satisfies multiple goals often requires that the hearer know about the speaker's plan, and reason about how the utterance fits into his plan. A speaker attempting to plan such an utterance must reason about what the hearer knows and how the hearer can interpret the speaker's intentions.

Consider the situation in Figure 1.1. The situation is typical of two agents cooperating on a task, where one has to make a request of the other. The speaker points to one of the tools on the table and says, "Use the wheelpuller to remove the flywheel." The hearer, who is observing the speaker while he makes the request, and knows that the speaker is pointing to a particular tool thinks to himself, "Ah, so that's a wheelpuller. I was wondering how I was going to get that flywheel off."

In this situation, the speaker's utterance affects the hearer far beyond a simple analysis of the propositional content of the utterance. Most obviously, the speaker is requesting the hearer to carry out a particular action, since the use of the imperative strongly suggests that a request is intended. However, the speaker includes using the wheelpuller as part of his request. If he knew that the hearer did not know that he was supposed to use the wheelpuller to remove the flywheel, then his utterance also serves to inform the hearer of what tool to use for the task. In addition, the

Figure 1.1

Satisfying Multiple Goals with a Request

fact that the speaker *points* to a particular object communicates his intention to refer to it with the noun phrase "the wheelpuller." Since the intention to refer has been communicated, the noun phrase also communicates the fact that the intended referent is a wheelpuller. The speaker could have just said, "Use *that thing* to remove the flywheel," if he had no goal of informing the hearer that the tool is a wheelpuller. (In fact, pointing may be the only way to successfully refer to an object where the only mutually believed description of it is that it is some sort of thing.)

1. Why a General Planning Mechanism is Needed

Figure 1.1 illustrates how understanding a speaker's physical actions can be important for understanding an utterance. The only reason that the noun phrase "the wheelpuller" informs the hearer is that the speaker has already communicated his intention to refer by his pointing action, relying on the speaker's knowing the connection between the physical act of pointing and the linguistic act of uttering a noun phrase. Since linguistic acts and other physical acts can be interpreted together in reasoning about a speaker's intentions, a language-generation system that treats physical and linguistic actions as uniformly as possible will enable the production of utterances that, like the one in Figure 1.1, satisfy multiple goals.

The example in Figure 1.2 provides additional evidence for the need to integrate physical actions and linguistic actions into a single planning system. In Figure 1.2 the agents are faced with a problem similar to that of Figure 1.1, but the agent making the request happens to be holding a box, which prevents him from pointing to the wheelpuller as he did in figure 1.1. If he says the same thing as he did in Figure 1.1 to realize his request, he will not succeed, because the hearer does not know what a wheelpuller is, and the speaker has not established his intention to refer, as he did in the previous example.

One option open to the speaker is to arrive at some description of the object that does not require pointing, and perhaps to inform the hearer that the object is a wheelpuller in a different utterance later. However, when there is no mutually believed basic-level descriptor (see Chapter VII), the resulting description will probably be awkward (e.g., "the thing with two arms and a large screw in the middle"). The speaker could also attempt to describe the object first and then refer to it,

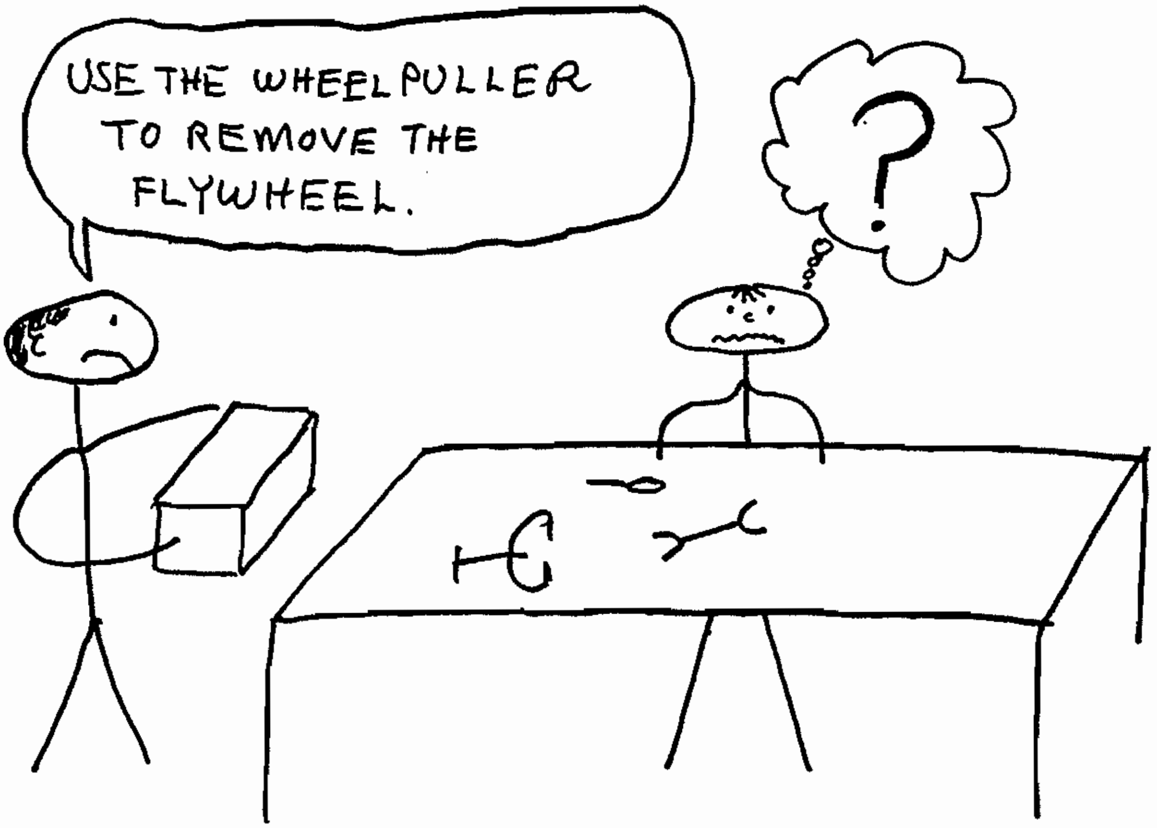

Figure 1.2

The Need for Integrating Physical and Linguistic Actions

however, this tactic can also be awkward. If an agent does not have physical actions at his disposal, then these techniques are the only alternatives.

Another alternative that could be planned when the speaker has both physical and linguistic actions at its disposal is for the speaker to set down the box, which would free his hands for pointing, and proceed as in Figure 1.1. As this example illustrates, relatively low-level linguistic planning, such as deciding what description to use to refer to something, can lead to the planning of physical actions. Such interaction provides support for the argument in favor of planning physical and linguistic actions uniformly.

A hypothesis of this thesis is that an agent's behavior is controlled by a general goal-satisfaction process. Agents are assumed to have goals that are satisfied by constructing plans from available actions. Given that an agent's overall behavior is controlled by such a planning process, it is advantageous for his linguistic behavior to be controlled by such a process as well. The reasons for this conclusion are (1) agents have to plan *both* physical and linguistic actions to achieve their goals, (2) linguistic and physical actions interact with each other, and (3) actions such as informing and requesting interact with each other and can be realized simultaneously in the same utterance. Since a language-generation system must reason about these interactions in order to produce natural-sounding utterances, a uniform process that plans both physical and linguistic actions is needed.

2. A Theory of Language Generation Based on Planning

Generating natural language by means of a general planning mechanism is a reasonable approach to the problem for a variety of reasons discussed in the previous section. However, this approach requires adopting a different view of language and communication than has usually been adopted in past language-generation research. Previous systems adopted a view of language processing analogous to that depicted in Figure 1.3, which illustrates a view that has been labeled the *conduit metaphor* by Reddy [80]. The conduit metaphor refers to the treatment of language as a pipeline or conduit that transfers information between the speaker and the hearer. The speaker has some idea of what he wants to say (represented by the semantic network inside his head), he encodes that idea in natural language (represented by the network wrapped inside the package), sends the package through the conduit

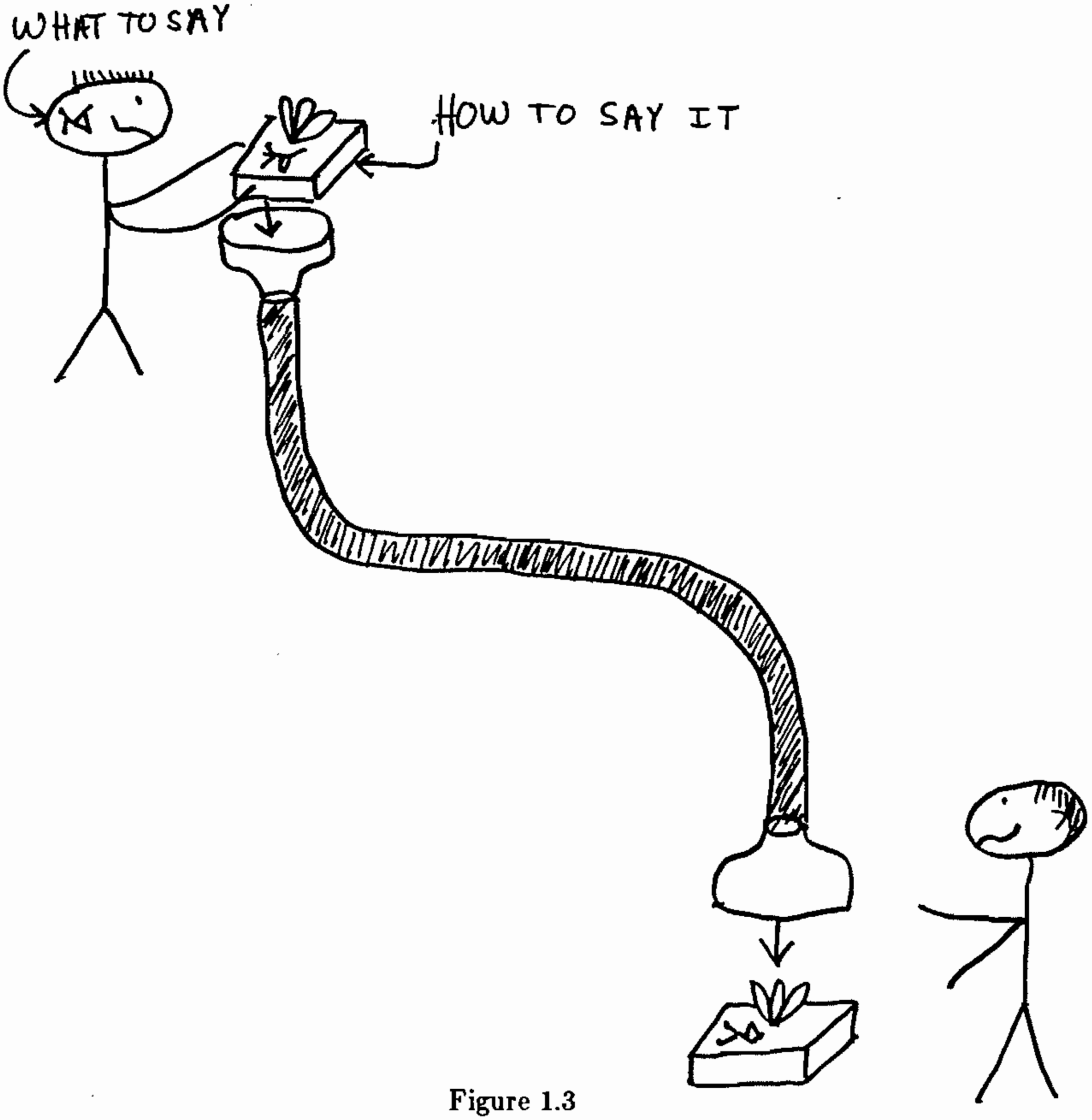

Figure 1.3

The Conduit Metaphor

to the hearer, who unwraps the package and removes the contents. This metaphor is quite pervasive in our common-sense intuitions about language and is reflected in many of our commonly used sayings, for example "He got his ideas across very well," or "He couldn't put his thoughts into words."

The disadvantage of this general view is that it forces one to acknowledge a very strong separation between two stages of the language-planning process: deciding what to say and deciding how to say it. The AI language-generation systems developed to date have focused primarily on the second of these two stages, and have assumed a role as a 'back-end' process of some other expert system. The expert system encodes what it wants to say in some internal data structure, and passes this structure to the generation module, which is supposed to decide how to encode it in natural language appropriate for the current context. The consequence of this separation is the inability of the language-generation process to have any influence on the behavior of the expert system, finding itself in a situation similar to the one depicted in Figure 1.2.

In contrast with the language-as-conduit approach outlined above, the approach advocated in this thesis (represented in Figure 1.4) treats language not as something to be transferred through a conduit, but rather as a set of actions available to agents that affect the mental states of other agents. This approach views decisions about 'what to say' and 'how to say it' as two phases of the same overall process, and recognizes the interactions between them. The design of an action appropriate for a given situation requires the consideration of a wide range of different kinds of goals that are satisfied by utterances, the knowledge of the hearer, general knowledge about the world, and the constraints imposed by the syntax of the language. The language planner can integrate these different knowledge sources to arrive at a plan involving abstract specifications of speech acts, and can finally produce English sentences. Instead of regarding the hearer as the mere consumer of a message, the language planner treats him as an active part of the communication process.

The planning system developed as a part of this research is called KAMP, which is an acronym for **K**nowledge **A**nd **M**odalities **P**lanner. KAMP is a hierarchical

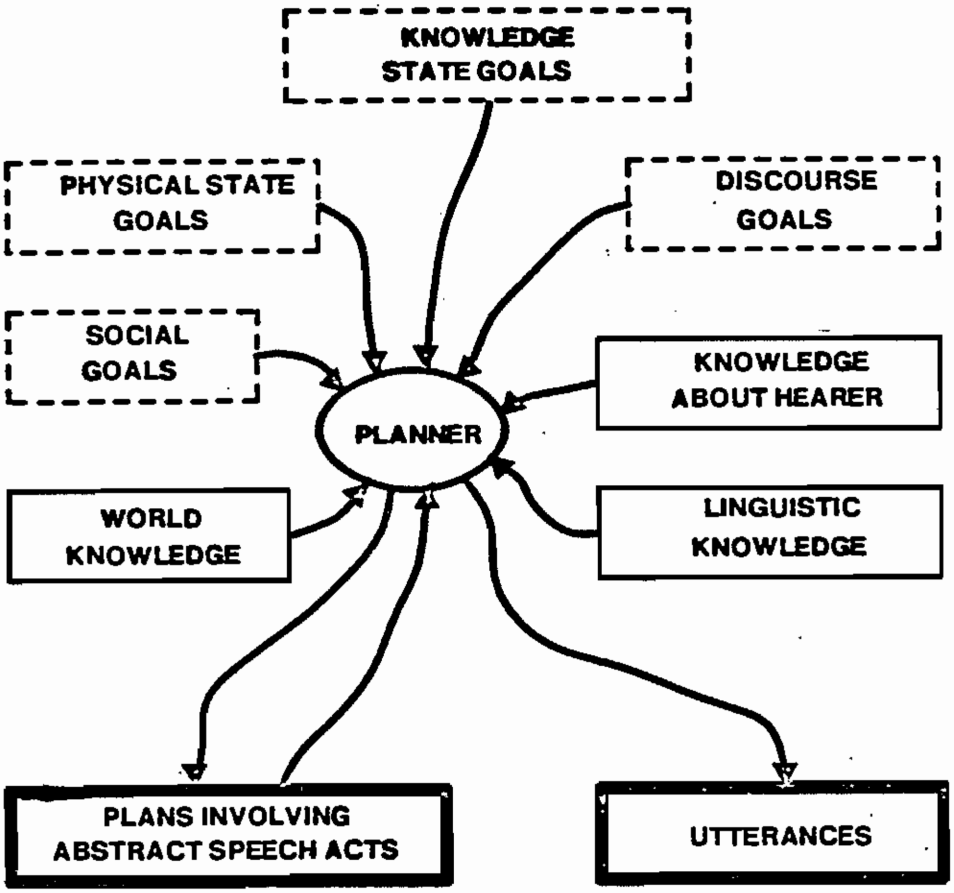

Figure 1.4

Overview of a Language Planner

planning system that uses a nonlinear representation of plans called a *procedural network* by Sacerdoti [86]. A hierarchical design for a language-planning system was selected because it provides for the separation between the planning of domain-level goals and actions and low-level linguistic actions, as well as for intermediate levels of abstraction that facilitate the integration of multiple goals into utterances.

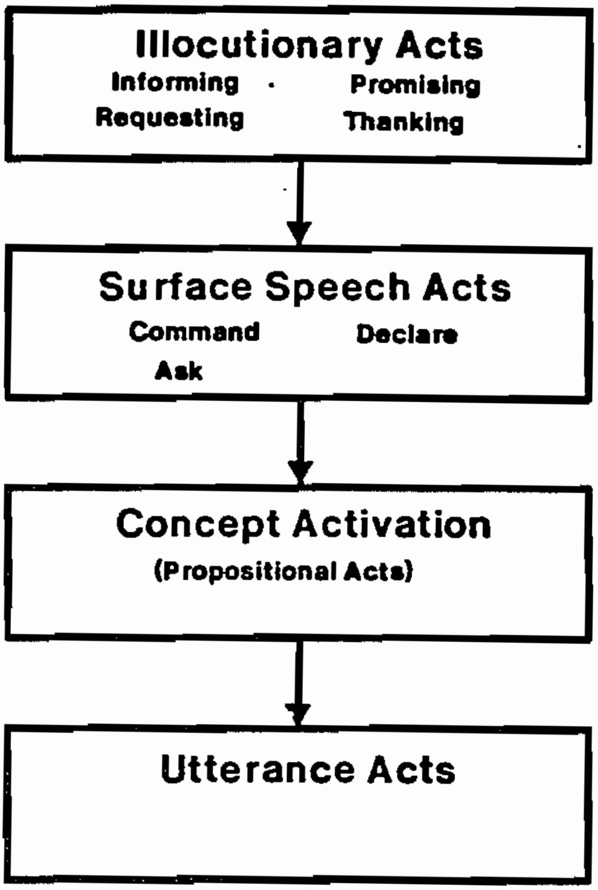

Figure 1.5

A Hierarchy of Actions Related to Language

The hierarchy of linguistic actions used by KAMP is represented in Figure 1.5. The planner can focus its attention on domain-level and high-level linguistic actions while ignoring details about choice of syntactic structure and descriptions for referring expressions. However, the uniformity of treatment of linguistic actions allows higher-level goals and actions to be influenced by the expansion of low-level linguistic actions. The mechanism KAMP uses to accomplish this is described in Chapter VI.

The highest-level linguistic actions are called *illocutionary acts*, which are speech acts such as informing or requesting represented at a very high level of abstraction, without any consideration for the action's ultimate linguistic realization. The next level consists of *surface speech acts*, which are abstract representations of sentences with particular syntactic structures. At this level, specific linguistic knowledge comes into play. One surface speech act can realize one or more illocutionary acts. The next level consists of *concept activation actions*, which entail the planning of descriptions that are mutually believed by the speaker and the hearer to refer to objects in the world. Concept activation actions are expanded as *utterance acts*, at which point, specific words and syntactic structures are chosen to realize the descriptors chosen for the concept activation actions. These syntactic structures have to be compatible with the sentential syntactic structure chosen when the surface speech act is planned. Concept activation actions can also be expanded partially as physical actions that establish the speaker's intention to refer, such as pointing. The detailed axiomatization and treatment by KAMP of each of these action type is described in detail in Chapters V, VI and VII.

3. An Overview of this Dissertation

This dissertation is divided into eight chapters, the first one of which you have almost finished reading. Chapter II reviews important related research in the areas of natural-language generation, planning and problem-solving, and philosophy and linguistics. Important ideas that have directly or indirectly influenced the development of the theory presented here are discussed. Chapter III is a detailed discussion of the possible-worlds-semantics approach to reasoning about knowledge, intention, and action. This chapter will contain familiar material if the reader is acquainted with Moore's approach [74] to reasoning about knowledge and action. Chapter IV describes the design of the KAMP multiple-agent planning system.

KAMP's general features for multiple-agent planning are described here without detailed reference to its language-planning abilities. The reader who is interested only in KAMP's application to distributed multiple-agent planning can read Chapters III and IV without reading the language-oriented chapters that follow them.

Chapter V describes the possible-worlds-semantics axiomatization of illocutionary acts in detail. The reader unfamiliar with Moore [74] should read Chapter III first. Chapter VI describes how KAMP plans surface linguistic acts, keeps track of the discourse focus, and plans concept-activation actions and indirect speech acts. Chapter VII describes a complete example of KAMP planning an utterance, starting with a high-level domain goal. Chapter VIII discusses the importance of the ideas in this thesis and potential avenues for future research that are opened up by this work.

II

AN OVERVIEW OF RELATED RESEARCH

0. Introduction

The planning of natural language utterances is an inherently interdisciplinary enterprise. Consequently, this thesis draws from and contributes to the state of the art in several areas, namely language generation, knowledge representation, planning, linguistics, and the philosophy of language, all of which draw upon recent results in cognitive science. In this chapter, work from these related fields that has had a significant impact on the development of the problem solving approach to language generation is reviewed.

1. Language Generation

A language planner is a language-generation system, and thus follows in the path of a number of earlier research efforts in artificial intelligence whose primary goal was the development of programs that would produce natural language effectively.

Several early language-generation systems, e.g., Friedman [26], were designed more for the purpose of testing a grammar than for communication. The earliest language-generation systems that were designed for communication depended upon ad-hoc strategies that produced reasonable behavior in predictable situations. An example of such a language-generation system was Winograd's SHRDLU [104]. SHRDLU produced language by having a large set of patterns with variables that could be instantiated appropriately in different instances. These patterns were combined with a number of heuristics about answering questions, referring to objects and pronominalization, and enabled the system to produce dialogs that sounded quite natural, given the simplicity of the techniques. Since it was possible to get such reasonable performance from the application of simple techniques, the problem of language-generation was considered less interesting and urgent than that of language understanding, and hence received much less attention from researchers. This simple approach has been followed in a number of more recent application-oriented AI systems that needed a graceful way of interacting with a user. The explanation component of MYCIN [89], and of Swartout's digitalis therapy advisor [98] are two examples.

In the early 1970s, some research was done to extend the simple approach of instantiating patterns to more general grammar-based approaches. These systems shared a reliance on a grammar of the language, usually expressed as an ATN, to embody the system's linguistic knowledge. The language-generation systems would accept an input in the author's favorite internal representation, and traverse an ATN, which would produce a natural language sentence as a result of the traversal.

One of the earliest of these grammar-based generation systems was that of Simmons and Slocum [94][95]. The generation system used an ATN grammar that performed a function quite similar to the inverse of the recognition process, which in their system [95] was also based on an ATN grammar. The language generator would be given a sentence in an internal representation, for which it would first select a verb to express the basic action or stative relationship, and would then pass the representation to an ATN. The generation ATN had tests on the various arcs that would query features in the input data structure, together with features of the chosen verb. The result of traversing an arc would be the production of a word, a clause, or a prepositional phrase. Simmons and Slocum used a set of "paraphrase rules" to relate synonymous lexical choices to the underlying semantic structure. These rules made it possible to generate both "Wellington defeated Napoleon at the Battle of Waterloo," and "Bonaparte lost the Battle of Waterloo to the Duke of Wellington." The question-answering algorithm used some very simple heuristics to match the lexical choices of the answer with those made by the user of the system in asking a question, which led to the adoption of the proper lexical choices much of the time. An example of such a heuristic would be *"Use the same verb in answering the question as the speaker did in asking it."* Such a heuristic would favor the generation of the second sentence above in response to the question "Who lost the Battle of Waterloo?" producing reasonable behavior without any analysis of how the sentence fit into the discourse.

Simmons and Slocum's system was another example of how far it is possible to go in language-generation using relatively simple techniques. However, their system had no notion of how an utterance fits into a discourse other than pattern matching against the user's question. As a result, it could perform only the simplest generation of definite references. Also, it was designed purely as a question-answering system that never took the initiative in a dialog.

Goldman [28] also developed an ATN-based language-generation system that focused on a different set of issues. Simmons and Slocum deliberately chose to have a large number of primitive concepts in their representation system, which simplifies the problem of lexical choice considerably. Goldman, for theoretical reasons, assumed a knowledge representation that was based on a very small number of predicates (see conceptual dependency in [87]). The primary problem that

Goldman addressed was that of finding a good lexical choice that would describe a concept that was encoded in the internal representation as relationships between a large number of semantic primitives. His solution was to use a discrimination net to filter possible lexical choices.

Goldman's generator was designed as part of the question answering component of the MARGIE system [87], and since it was designed as a question answerer that produced responses with only the question for a discourse context, it suffered from most of the deficiencies of the Simmons and Slocum generator.

The systems of Goldman and Simmons and Slocum are the paradigms for a number of language-generation systems developed subsequently, including Wong's semantic network language-generation system [107], and the generation component of the HAM-RPM system [35]. A generation system called PENMAN has been developed by Matthiessen [61] based on a systemic grammar that generates English sentences from fragments of KL-ONE nets [7].

McDonald [67], [68] has developed a generation system called MUMBLE that differs significantly from either the pattern instantiation or the ATNgrammar-based approaches. MUMBLE probably has the broadest coverage of the English language of any generation system developed to date. McDonald adopted the hypothesis that the best design for a language-generation system should reflect in its performance certain observations about human language production. Although the system was not constructed specifically as a psycholinguistic model, it embodies many assumptions about human language production that are used to computational advantage in the system. For example, decisions about the realization of a message element cannot be retracted once they have been made. McDonald claims that human language production conforms to a similar determinism principle, and that conformation to this principle has the advantage of limiting the amount of processing that needs to be done to produce an utterance.

McDonald separates the language-generation process into three levels. The highest level is the "expert system". The expert system knows about problem solving in some domain, but does not necessarily have to know anything about language. The lowest level (and the level realized by MUMBLE) is the "linguistic component", which knows about English grammar, has a lexicon appropriate to the application domain, and processes some information about the intended audience. McDonald also proposes an intermediate level called the "speaker component", which acts as an interface between the expert system and the linguistic component. The speaker component knows what the expert system wants to say, knows what kinds of data structures are expected by the linguistic component, and encodes an appropriate message to be passed to the linguistic component for generation.

The language-generation process is a two-phase process. The first phase expands the message into a tree representing the surface syntactic structure of the utterance. The second process traverses the tree built by the first process, printing words, annotating the grammatical context, recording the history of the process, and propagating grammatical constraints.

The majority of the work in MUMBLE is done by procedurally encoded rules in the grammar and lexicon. These procedures, which are invoked by the controller at appropriate times while traversing the syntactic structure tree under construction, figure out what the best realization of a particular message element is within its context in the tree, and test conditions in the discourse state, audience model, etc. to determine which options to take in making decisions about, for example, pronominalization and choosing between different syntactic structures.

Although MUMBLE's coverage of the English language is broader than other generation systems and attends to some discourse phenomena (e.g., it has reasonable heuristics for pronominalization and definite reference), it has some limitations. Since it has the advantage of being portable between different expert systems, which may use different knowledge representations, it has the disadvantage of not being able to reason about a world model. The effect of reasoning with a world model is obtained when the system implementor writes grammar routines that are capable of invoking the expert system's knowledge in their decision procedures. These decision procedures must be implemented for each domain, and although the system can, in principle, do some of the kinds of planning mentioned in this thesis, it cannot do it in the same general, domain-independent manner. McDonald has not described the audience model component in detail, so it is not clear what its limitations are. An important limitation arises from making a distinction between "what to say" and "how to say it" at a very high level. It becomes difficult under such circumstances for linguistic planning to have much influence on the agent's overall plans and to integrate linguistic and nonlinguistic actions.

Some current research projects are investigating other issues in language generation. One such project in progress is that of McKeown [69], who is concentrating on the problem of generating multisentence responses to queries about a data base schema. McKeown's basic approach is to define a number of organizational schemata such as *compare and contrast, illustration by example,* and *analogy,* and use rules associated with each schema to incorporate relevant information into a coherent text. Mann and Moore [60], [72] have also done some work in organizing a large body of knowledge into coherent text by dividing the problem into a sequence of problem-solving stages that deal with the problem at different levels of abstraction. They have developed a system called KDS that embodies this theory.

Gabriel [27] developed a language-generation system called YACKETY-HACKS that explores some interesting issues in the design of control structures for what he calls *fluid domains,* of which he claims natural-language generation is an instance. Gabriel's claim is that language production can be done on a number of different levels by processes of varying levels of generality, with a potentially large number of knowledge sources capable of contributing relevant information. YACKETY-HACKS is essentially a control mechanism for integrating these knowledge sources.

2. Goals, Plans, and their Influence on Utterances

The early work in the field of artificial intelligence on the relationship of plans and goals to language was not done in the area of generation, but rather in the area of understanding. Bruce [9] did some of the early work that set the stage for true speech-act planning. He started from the viewpoint that language is purposeful behavior, and the task of understanding a sentence is not only a process of recovering the meaning, but also of interpreting the speaker's intentions behind producing the utterance. Bruce proposed developing a computational formalism for representing an agent's beliefs and for describing actions such as speech acts that affect beliefs. The formalism was never developed to the point where it would be implementable, and it was never realized in a working system, but the basic direction taken in his research was important.

Recent work in the understanding of simple stories [88] has recognized the need for reasoning about the underlying intentions of agents in order to understand stories about them. Much early work on the understanding of stories relied on matching events in the story with some stereotypical sequence of actions called a "script". It was soon realized that it was impossible to capture every possible sequence of events beforehand, and that some general mechanism of understanding the plans of the agents involved in the story was essential. A model was developed in which agents could plan various actions including asking and telling. Although much of this work lacked the formal rigor that could make it the basis of a language planning system, it was nevertheless a step in the right direction. The model was used to understand stories about agents achieving their goals [102].

Meehan [88] extended these early ideas about planning to the design of a system that produced simple stories. Meehan's system was not a language-generation program since it did not produce any actual English sentences. What it did was to compose formal descriptions of short stories about different agents who would make plans to achieve their goals, and could be frustrated by various situations and events. The agent's plans included actions of telling, asking, and persuading.

Kaplan [47] designed a data base question answering system that would attempt to provide helpful responses to a user. For example, if the answer to a query such as *"How many students in CS 243 received a grade of 'A'?"* was zero because CS 243 was not offered that quarter, the system would recognize that the query failed because of a presupposition failure, and would reply that CS 243 was not offered, instead of simply answering none. The system made some simple assumptions about what the speaker's intentions behind asking a question were, and without trying to do a great deal of sophisticated planning, attempted to provide the response that was most appropriate for the user's plan.

Allen, Cohen and Perrault have done considerable work in extending the ideas of Bruce [9] by developing implementable formalisms that were incorporated in working systems for the planning and recognition of speech acts [1][2][15][16]. Allen

[1] designed a system that would understand indirect speech acts by attempting to recognize the speaker's plan, and trying to see how the utterance could fit into that plan. Cohen [15] is concerned with the problem of producing an appropriate speech act to satisfy a speaker's goal. Cohen implemented a system called OSCAR that can plan for a hearer to recognize the speaker's intention to perform a speech act, and thereby succeed at informing him or requesting something of him. Cohen's system produces a specification of the speech act, naming the type of action to be performed, the agent, and the propositional content of the act, but it does not actually produce English sentences.

The utterance planning undertaken so far in this thesis, as well as other speech act planning work such as that of Allen, Cohen, and Perrault, works in domains that are fundamentally task oriented, as in performing some cooperative problem solving task, or assisting a customer at an information booth. This work leaves open the question as to whether planning and problem solving techniques are also useful in less well-structured domains. Hobbs and Evans [45] examine the goal structure in a "small talk" dialog and conclude that in fact they are. The goals that arise are of a different nature — more social goals are involved for which formal description is difficult, however it is clear that similar principles operate in the more loosely structured domains as well.

3. Planning, Problem Solving, and Knowledge Representation

Since this work is about *planning* utterances, this review would not be complete without acknowledging the debt owed to previous research efforts in planning and problem solving. The planning system described in Chapter IV builds on ideas embodied in the early systems described below.

STRIPS [23] was one of the first planning systems. It can be characterized as using a first order logic description of states of the world, with an extra-logical set of operators with add and delete lists that transform one state into another. The basic control strategy was backward chaining from a partially specified goal state.

A number of procedural planning languages (e.g., PLANNER [42]) were developed that were similar to STRIPS in that they used a data base of assertions to represent knowledge about the world, and extra-logical operations that added and deleted assertions in the data base. The difference lay in the encoding of planning operators as procedures that would perform manipulations on the data base as a side effect of their execution, and in a closed-world assumption that was strongly built into their operation. They allowed control structures that included both forward and backward chaining.

Kowalski [50] demonstrated that it was not only possible to formalize planning entirely within logic, but that with appropriate constraints on the axioms, planning could be carried out by normal deduction procedures with about the same complexity as the STRIPS approach.

The next major advance in planning was the encoding of planning operators in a hierarchy of abstraction, as advocated by Sacerdoti [86]. It sounded intuitively desirable and at least possible on inspection that the space the planner had to search could be significantly reduced if it could form a rough cut plan first using abstract operators, and later refine the rough plan into a more concrete low-level plan. "Critic" procedures would be employed to resolve what would hopefully be minor inconsistencies arising at the lower level.

Of course there is no guarantee that the structure of the high-level plan would look anything at all like the final low-level plan, and this approach would only work for problems that were nearly decomposable. It is strictly an empirical fact that this property holds for a large number of planning domains. In spite of this shortcoming, it appears that hierarchical planning has wide applicability in many areas, including language planning. Much current planning research deals with the problems that arise in circumventing interactions that take place between actions in a plan and involve the choice of instantiations for variables in the plan. Such ideas as Stefik's "constraint posting" [96] and Hayes-Roth's "opportunistic planning" [40] are attempts to solve some of these problems. A good review of different robot planning systems and the types of problems they can and cannot handle can be found in Nilsson [78].

The knowledge representation used by the language planning system for reasoning about what agents know owes much to the research of Moore [74]. Before Moore, most systems that had to reason about propositional attitudes did so with over simplistic and ad-hoc techniques, since solving such reasoning problems were not the primary goals of the research. Moore's work on a possible-worlds-semantics approach toward reasoning about knowledge and belief is the first body of work to be directed primarily toward that end. This work and some alternative related approaches are summarized in detail in Chapter III.

4. Philosophy, Psychology, and Linguistics

The work of philosophers, psychologists, and linguists has a different orientation from the work reported in this thesis. As research in the field of artificial intelligence, this thesis has the goal of establishing the viability of a theory of language production that can be used computationally. The computational embodiment of a theory of language is the subject of this thesis, but the theory itself owes its origins to previous work in other fields.

The view of language production as a planning process owes much to the development of speech act theory as developed by Austin and Searle [5], [90], [91], which views utterances as actions performed by speakers to achieve intended effects. Searle attempted to elaborate on this view by specifying explicit preconditions and effects for different types of speech acts. One of Searle's most important contributions was the establishment of the importance of recognition of intention in the production and the understanding of speech acts. This work is discussed in greater detail in Chapter V.

Chafe [11] examines pauses in natural narratives to describe the processes that people use to produce utterances. The focus of that work is to discover what the speaker's processing strategy can reveal about the organization of memory. He proposes a hierarchical memory organization composed of *memories, episodes, thoughts,* and *foci,* and proposes that pauses in utterances correspond to transitions between these different units of information storage. This work provides evidence about how people organize bodies of knowledge into coherent text, and how they recover from planning errors and false starts, which is useful for developing a plan-based theory of language production. Also, this research provides a basis on which the psychological plausibility of a computational system such as the one suggested here can be judged.

Levy [56] uses concepts of communicative goals and strategies to develop a framework for analyzing naturally occurring spoken discourse. He extends this formulation [57] and proposes for the production of text a "production model" and an "artifact thesis" that tie together many previous efforts in different disciplines to describe discourse. The research reported in this thesis may be seen in part as an effort to formalize the integration of some of the multiple perspectives on an utterance described by Levy.

The idea that utterances are part of a speaker's plans to achieve his goals has appeared in the linguistics literature under different guises for quite some time. A number of modern linguists have looked at language beyond its properties as a formal symbol system, and have examined questions of how language is used and evolves within a sociocultural setting to serve a variety of functions. Halliday [38] has advocated breaking away from a view of language exclusively as an information conduit, and emphasizes the importance of all the functions of language and how a speaker uses it in various settings. Morgan [76] argues for an event-action based view of language as opposed to what he calls the object view which treats utterances as formal objects. This distinction is very much within the spirit of this thesis. Linguists who have worked within speech act theory (e.g., Cole, Gordon, Grice, Lakoff, and Morgan, to name a few) have established a theoretical foundation for the linguistic part of language planning and have collected much empirical data that provides a set of phenomena against which to test the adequacy of a plan-based theory of language.

III

REPRESENTING KNOWLEDGE ABOUT INTENSIONAL CONCEPTS

0. Introduction

This chapter examines some of the special requirements of a knowledge representation formalism that arise from the planning of linguistic actions. Language planning requires the ability to reason about a wide variety of *intensional concepts* that include knowledge, mutual knowledge, and belief. Intensional concepts can be represented in intensional logic by operators that apply to both individuals and sentences. What makes intensional operators different from ordinary extensional ones such as "$\wedge$" and "$\vee$" is that one cannot substitute terms with the same truth value inside the scope of one of these operators without sometimes changing the truth value of the entire sentence. Planning linguistic actions requires a uniform formalism for representing all the different intensional concepts because the different concepts are interrelated and therefore interact during the course of solving a single problem. For example, for an agent A to plan a request of B, A must reason about how B's *knowledge* of A's *wants* affects B's *wants*.

This chapter describes a knowledge representation based on a possible-worlds semantics for a modal logic that is adequate for representing the knowledge needed by a cooperative agent to participate in task-oriented dialogs, and is capable of being used in an efficient manner by existing first-order-logic deduction systems. This possible-worlds semantics approach and its integration into a first order logic deduction system was developed by Moore [74]. I have made some extensions to deal with cases of wanting and mutual knowledge not considered by Moore; otherwise I have adopted Moore's approach essentially intact. Since an understanding of the overall approach is necessary to understand the KAMP language planning system and the axiomatization of illocutionary acts, both Moore's basic approach and the extensions that have been adopted are described in this chapter.

Each of the concepts of knowledge, belief, and intention discussed in this chapter has provided fuel for centuries of philosophical debate. It is not my intention to settle issues such as when true belief constitutes knowledge, or even to advance an opinion on them. Moore's representation is neutral with respect to most of these issues. However, the representation is intended to provide sufficient generality and flexibility so that the designer of a system using the representation can adopt whatever philosophical perspective on these issues he deems appropriate for the situation. This thesis takes a pragmatic approach to most of these issues, making assumptions that lead to the simplest system that behaves reasonably in task-oriented dialogs.

Much attention in the literature of artificial intelligence has been devoted to problems of representing knowledge about the world. Many formalisms have been proposed, and those that had a coherent enough semantics to be expressible in first-order logic were for the most part merely syntactic variations or a proper subset of first-order logic. Although many of these representation systems are designed to address substantive issues in memory organization and control of deductive processes, their representational power has usually been weaker than that of full first-order logic.*

The view adopted in this chapter is that the central problem of knowledge representation is that of finding an appropriate set of axioms to express facts about the world and to draw the appropriate inferences from them. To that end, I will express all the axioms in the notation of standard first-order predicate calculus with functions and equality. For the sake of notational clarity, some axioms may also be expressed in a modal language of knowledge, action, and wanting for which the semantics is readily expressible in first-order logic. Although there may be substantial issues involving the efficient storage and retrieval of the axioms, such considerations are beyond the scope of this investigation.

A number of arguments have been advanced from time to time of the form "Logic (or logic of type X) cannot be used for reasoning about natural language because the semantics of sentence A is sentence S in the logic, and S clearly does not represent the intentions of the speaker." An example of this type of such an argument is that the semantics of the sentence "John knows where Sam lives" is $\exists x\,\mathbf{Know}(\mathrm{John}, \mathrm{Abode}(\mathrm{Sam}) = x)$ and this must be wrong because it doesn't account for the fact that in one case (for example in taking a census) I may say "John knows where Sam lives" if all John knows is what city John dwells in, and if I were going to Sam's party, I would not agree with the statement.

The fallacy in this argument and many similar ones is the attempt to too closely identify the meaning of a sentence in natural language and the meaning of a sentence in the logic which is the result of a simplistic semantic analysis. Winograd [106],

---

* There are few examples of representation systems that are capable of representing facts that are unexpressible in first-order logic. Systems designed for nonmonotonic reasoning ([6], [63], [65], [82],) and systems that attempt to represent and reason with uncertain knowledge ([39], [89], [99]) are the few exceptions.

for example, presents some convincing arguments that semantics of natural languge cannot be governed by straightforward compositional rules. This thesis encourages the adoption of logic as a tool for representation and as a formal model for reasoning processes. This certainly does not make any claims about the process by which natural language utterances are related to this formal model. Indeed, one of the key observations of this thesis is that the "meaning" of an utterance (defined as what the speaker intends the hearer to realize the speaker is trying to do by means of producing the utterance) is intimately associated with the respective beliefs and wants of the speaker and hearer and the state of the discourse at the time of the utterance. Although all these concepts may be expressible in the logic, there is no simple sentence in the logic that one could describe as "the meaning of the sentence" in isolation from all the previously mentioned influences. It makes sense to talk about sentences having a logical form, so a sentence like "John knows where Sam lives" could have the logical form

$$\exists x\, \mathbf{Know}(\mathrm{John}, \mathrm{Abode}(\mathrm{Sam}) = x),$$

but what the effect of uttering such a sentence at a particular time and place for a given hearer is another consideration not captured entirely by the logical form.

1. Modal Logic and Possible Worlds Semantics

It is quite natural to represent intensional concepts as sentential operators in a modal logic. This representation gives one the ability to write statements that express the relation between the scopes of the intensional operators and quantifiers, such as

$$\mathbf{Know}(\mathrm{John}, \exists x\, \mathbf{Want}(\mathrm{Bill}, P(x)))$$

which is taken to mean that John knows that there is some particular thing that Bill wants to have property P (but John does not necessarily know what that thing is). This can be distinguished from

$$\mathbf{Know}(\text{John}, \mathbf{Want}(\text{Bill}, \exists x\, P(x)))$$

which means John knows that Bill wants there to be something with property P, and from

$$\exists x\, \mathbf{Know}(\text{John}, \mathbf{Want}(\text{Bill}, P(x)))$$

which means there is some particular thing known to John, and moreover, John knows Bill wants it to have property P.

The sentential operators of possibility and necessity in traditional modal logics are a bit different from **Know**, since **Know** can operate on individuals, as well as sentences. However, as Moore points out, the logic of knowledge is quite similar to the standard modal logic S4 with **Know** being equivalent to necessity, and if the knower is held constant, the logic really is S4, so one is justified in calling **Know** a modal operator.[*] New intensional operators will be freely introduced into the logic where appropriate, with appropriately defined semantics.

Unfortunately, no efficient automatic deduction techniques for reasoning in quantified modal or intensional logics have yet been developed.[**] What would be ideal is to reduce statements in the intensional logic to first-order logic, and do the reasoning in first-order logic, since many first-order logic deduction systems

---

[*] For a discussion of modal logic and a definition of S4, see Hughes and Cresswell, [46].

[**] Konolige [49] has recently developed a resolution proof procedure for a restricted class of modal logics that are useful for reasoning about knowledge.

currently exist, and the problems of reasoning are better understood there than in other types of logic.

Kripke [51], [52] developed the idea of a model theory for modal logic that is based on possible worlds. A possible world is a formal object that can intuitively be thought of as representing a state of affairs that might actually have been the case. The possible worlds semantics for the standard modal operators of possiblity and necessity is easy to describe. A proposition is *possible* if it is true in some possible world. Similarly, a proposition is *necessary* if it is true in every possible world. Logicians have noticed that there are a number of cases that are not covered by these axioms, for example, propositions that are true in all possible worlds may or may not be necessarily true in all of them, or a proposition that is possibly true may or may not be necessarily possible. These observations gave rise to a proliferation of modal logics, each with axioms to cover these possibilities in a different way.

Kripke proposed that one regard possible worlds as not being absolutely possible, but only as being possible relative to some other world. Kripke defined *accessibility relations* on the possible worlds that described explicitly which worlds are possible alternatives to which others, and then proved that all the modal logics could be unified with the same semantics, with only different accessibility relations in each case.

Hintikka [43], [44] developed a modal logic of knowledge and belief with a semantics that was closely related to Kripke's possible worlds semantics. This approach was adopted and extended by Moore [74] for reasoning about knowledge and action and is further extended in this thesis to cover the concepts of mutual knowledge and wanting necessary for the planning of illocutionary acts.

2. Representing Knowledge about Knowledge

The key to developing any Kripke-like semantics for an intensional logic is to define the meaning of the sentential modal operators in terms of accessibility relations between possible worlds. One can then axiomatize the properties of the accessibility relations in first-order logic, and instead of reasoning about the truth of propositions, one can reason about the relations that hold between different possible worlds. Adopting the latter approach, the designer of an AI system can cast the entire axiomatization of the world in first-order logic, and bring his well developed set of deduction tools to bear on solving the problems in his domain.

The approach to reasoning in a modal logic by reasoning about its semantics may seem counter-intuitive at first. Many axioms for defining intensional operators tend to be obscure. This obscurity does not make it any easier to arrive at the semantics for an intensional operator. There is no magic method to tell what the right possible-worlds semantics for a modal operator is. One must rely on one's intuitions about the common-sense concept that the intensional operator is intended to capture, and decide if the proposed formal semantics agree with those intuitions in the most critical areas. This criterion renders irrelevant any considerations about whether possible worlds "really exist" in some sense or have psychological reality. Possible-worlds semantics is just a formal tool for modeling certain kinds of inferences that people make and that are desirable for an AI system to make also.

In many cases, a simple axiomatization will lead one to draw conclusions that are intuitively implausible. One of the most obvious deficiencies in the formalism presented here is that it forces a knowledge closure assumption: every agent knows all logical consequences of his knowledge. Certainly no one would claim that this is actually the case, however no one has yet proposed an adequate means to characterize precisely what inferences an agent is capable of making or not making. Moore [74] has suggested treating deductions involving another agent's application of modus ponens as merely a plausible inference that would be the first candidate for retraction when an inconsistency arises. It is not clear that the closed-knowledge assumption is a serious shortcoming, since in many problem-solving applications the knowledge closure assumption works quite well. The problem of knowing all consequences of an agent's knowledge can be regarded as a mildly unfortunate side effect of a formalism that gives one a great deal of power to reason about language. The formalism presented here has a number of advantages over others proposed to solve many of the same problems. However, since the focus of this thesis is on language production, the deficiencies of the representation when pushed to its limits will be acknowledged and deferred for further research.

The approach that will be adopted is the stating of basic facts about the world in an intensional *object-language* that is translated into a first-order *meta-language*. A set of axiom schemata serve as translation rules that describe the relationship between the two languages. The object-language is an intensional language that talks about objects, relations and actions in the physical world, and the mental states of agents. The object-language has all the quantifiers and logical connectives of an ordinary first-order theory, except that they are of a different nature. Logical connectives such as "$\wedge$" and "$\vee$" are actually functions in the object language that map object-language formulas into other intensional objects. However, since the translation is very straightforward because the connectives behave like their corresponding equivalents in the meta-language (e.g., $\forall w\, T(w, P \wedge Q) \equiv T(w, P) \wedge T(w, Q)$), I will use the same symbols for both the object-language and meta-language. The treatment of quantifiers is somewhat more difficult, and is explained in detail.

The meta-language is an extensional language that has as its domain of discourse individuals, relations and actions, and in addition, possible worlds and all well-formed expressions in the object language. The meta-language also has predicates for describing the accessibility relations between possible worlds. Some intensional operators may be realized directly in terms of accessibility relations (such as **Know**) and others may be realized indirectly. Thus, the object-language can be regarded as a "high-level" language that is "compiled" into the meta-language using translation axioms that relate the object-language to statements in the meta-language about possible worlds. In this thesis, object-language intensional operators will always appear in **boldface** roman type, predicates will appear in *UPPER-CASE* italic type and functions and constants in Lower-case roman type with an initial capital. Meta-language variables are in *lower-case* italic type and object-language variables are in *lower-case* italic type preceded by an initial '?'. Most of the notational conventions and predicate names are taken directly from Moore [74] to facilitate cross reference by the reader desiring additional information.

The first task of axiomatizing the semantics of an intensional logic is to devise a formal method for stating that a proposition is true in a possible world. The basic axioms about the semantics of knowledge are the same as described by Moore in [74]. A meta-language predicate, T, which applies to a possible world and an object-language expression, is used to describe this relationship. One possible world is distinguished by virtue of being the current real world, designated W_0. A statement in the object-language is true if and only if it is true in W_0. The intensional operator,

True, is introduced into the meta-language to express that the object-language expression is true in the real world.*

$$\textbf{True}(P) \equiv T(W_0, P)$$

Thus, one way to represent the fact that "Reagan is president" is true would be

$$\textbf{True}(EQ(\text{President}(\text{Usa}), \text{Reagan}))$$

which is equivalent to

$$T(W_0, EQ(\text{President}(\text{Usa}), \text{Reagan})).$$

The next task is to define an accessibility relation on possible worlds that characterizes the semantics of **Know**. We will say that it is true in some possible world w that an agent A knows that a proposition P is true if, for every possible world accessible from w, given the knowledge accessibility relation K for the agent A, P is true in that world. In the meta-language, this is expressed by the following axiom:*

$$\forall w_1\, T(w_1, \textbf{Know}(A, P)) \equiv \forall w_2\, K(A, w_1, w_2) \supset T(w_2, P) \qquad (K1)$$

What this statement means intuitively is that the only worlds that are possible as far as A is concerned are those that are consistent with what he knows. Since

---

* One may worry about problems of inconsistency when a language is allowed to talk about the truth of its own sentences (see Rogers [84], pp. 210–215) but as Moore points out in [74], as long as the language is restricted in such a way as to prohibit the construction of sentences that assert their own falsehood no problem will arise. Since the meta-language does not include itself in its domain of discourse, it cannot describe its own truth conditions, and hence, no paradox can arise.

---

* Axiom *(K1)* is not strictly correct, because it says nothing about the denotation of A with respect to $w1$ in the meta-language. Occasionally some details may be omitted in the interest of notational clarity and to not burden the reader with excessive detail on topics that have not yet been introduced.

he knows that P, P must be true in every possible world compatible with his knowledge. For example, at this moment I do not know whether Ronald Reagan is standing up or sitting down. Therefore the proposition "Reagan is standing" is true in some possible worlds compatible with my knowledge, and false in others. On the other hand, the proposition "Reagan is president" is true in *every* possible world compatible with my knowledge. The relation between possible worlds and a known proposition can be expressed by the diagram in Figure 3.1. In that figure, A knows P because P is true in every world related to W_0 by the accessibility relation K_A. A does not know Q, because Q is true in some accessible worlds and false in others.

The semantics of **Know** requires further elaboration to insure that it is possible to make all the inferences that could be regarded as intuitively plausible. An inference one frequently wants to make is that when someone knows something, then it is true. Bypassing a host of philosophical problems, this fact will be taken to distinguish knowledge from mere belief. This inference can be made by attributing a reflexive property to the K relation:

$$\forall A, w\, K(A, w, w). \tag{$K2$}$$

If it is not immediately obvious that reflexivity captures this fact, consider that what the reflexivity property says is that whatever world an agent is in is consistent with the agent's knowledge, and as a special case, the real world is consistent with any agent's knowledge. In other words, what actually *is* the case is possible according to what one knows. It is not difficult to infer that $T(W_0, P)$ from $\mathbf{True}(\mathbf{Know}(A, P))$, and axioms $(K1)$ and $(K2)$.

It is probably worth pointing out at this time that this formalization of knowledge

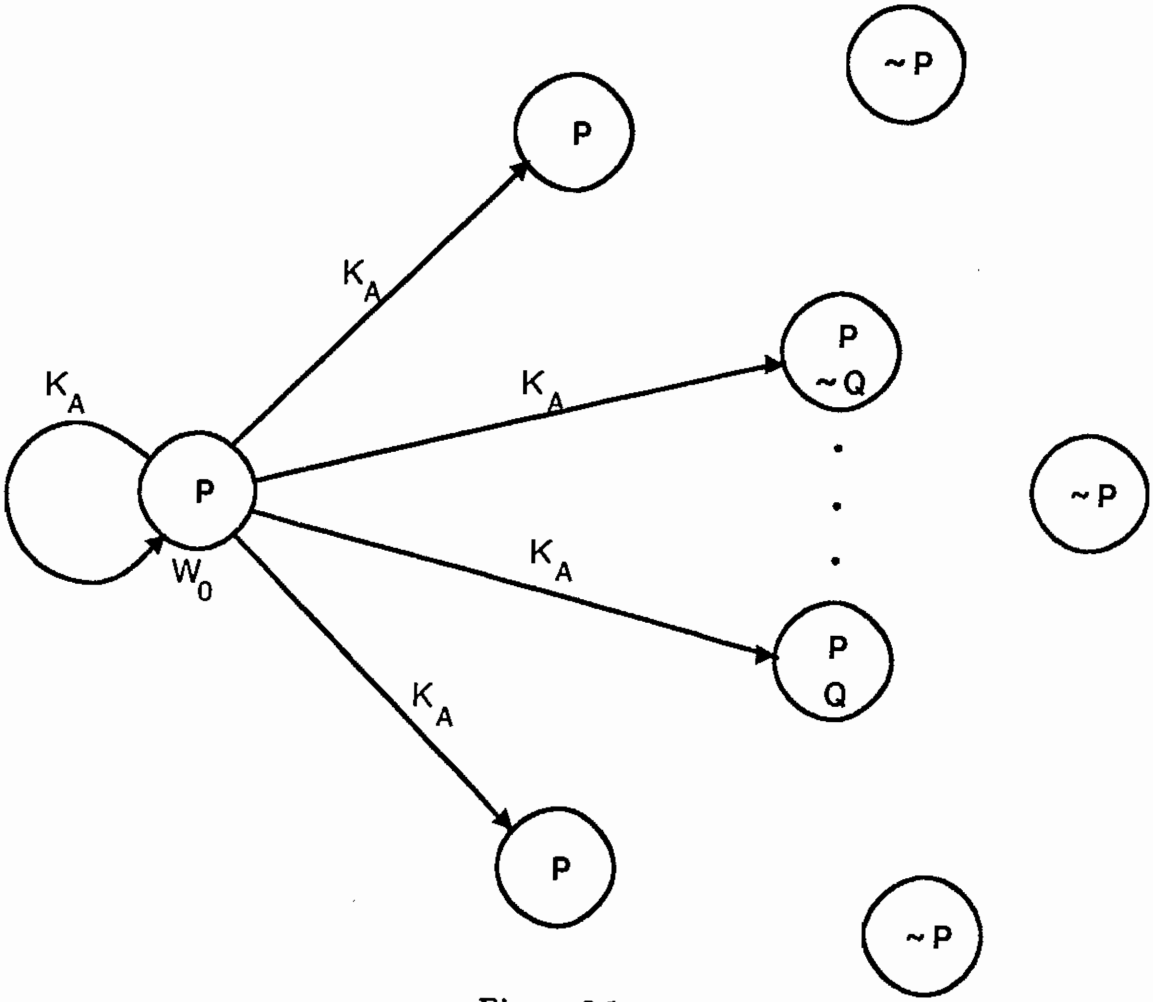

Figure 3.1

A Knows that *P* is True and *A* Does Not Know whether *Q*

makes a fairly strong distinction between knowledge and belief. It is impossible to *know* propositions that are not actually true. Of course no one would dispute the fact that in ordinary discourse we use the English verb "know" in a much broader sense in which it is perfectly proper to say something like "Back in July of 1980 I knew Reagan would win the election." The reason for narrowing our attention to the more restrictive definition of know given here is to avoid the multitude of extremely difficult problems that arise when attempting to consider beliefs that may not actually be true. We are faced with the problem of representing the fact that beliefs may be held with varying degrees of certainty, and that these certainties can change with the acquisition of new information. Since changes in the certainty of one belief can have an almost arbitrary influence on the certainty of any other belief held by the agent, the problem of maintaining consistency of belief is very difficult. Some work on truth maintainance systems (e.g., [22]) is relevant to this problem, but it is possible to address a number of interesting language problems without assuming the additional burden of belief revision.

Another inference that one frequently wishes to make is that if an agent knows something, then he knows that he knows it. To express knowledge about knowledge, we follow the same course charted so far, i.e., to state that A knows that he knows P is equivalent to stating that $\mathbf{Know}(A, P)$ is true in all worlds compatible with what A knows in the real world. This means that P is true in every world compatible with each world compatible with A's knowledge in the real world. This situation of knowing what one knows is essentially one of transitivity of the K relation and it is expressed in Figure 3.2 and in the following axiom:

$$\forall w_1, w_2, w_3\, K(A, w_1, w_2) \supset [K(A, w_2, w_3) \supset K(A, w_1, w_3)]. \qquad (K3)$$

It is easy to see that rules $(K1)$ and $(K2)$ can also be used together to prove the implication of Figure 3.2 in the other direction, i.e.,

$$\mathbf{Know}(A, \mathbf{Know}(A, P)) \supset \mathbf{Know}(A, P).$$

The semantics of knowledge about knowledge can be illuminated further by examining Figure 3.3, which shows the relation between possible worlds describing the situation of John knowing that Bill knows whether P is true, but not knowing himself whether P is true. This is a situation that a representation used by a

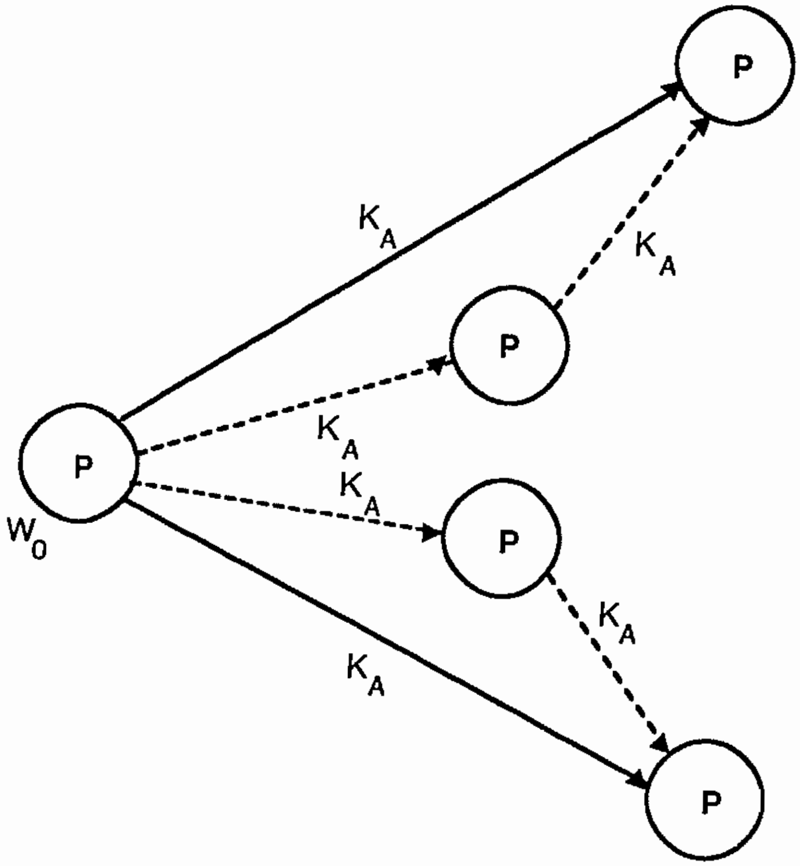

Figure 3.2

If A knows P, then he knows that he knows P.

language-planner must be capable of describing, but that many of the simpler proposed representations do not handle adequately. Such knowledge is needed to plan a question and decide who knows the answer so the planner will know whom to ask. In Figure 3.3, both P and $\sim P$ are true in possible worlds compatible with John's knowledge, so John does not know whether P. However, in all the worlds compatible with John's knowledge in which P is true, P is true according to Bill's knowledge, and in all worlds in which $\sim P$ is true, $\sim P$ is true according to Bill's knowledge.

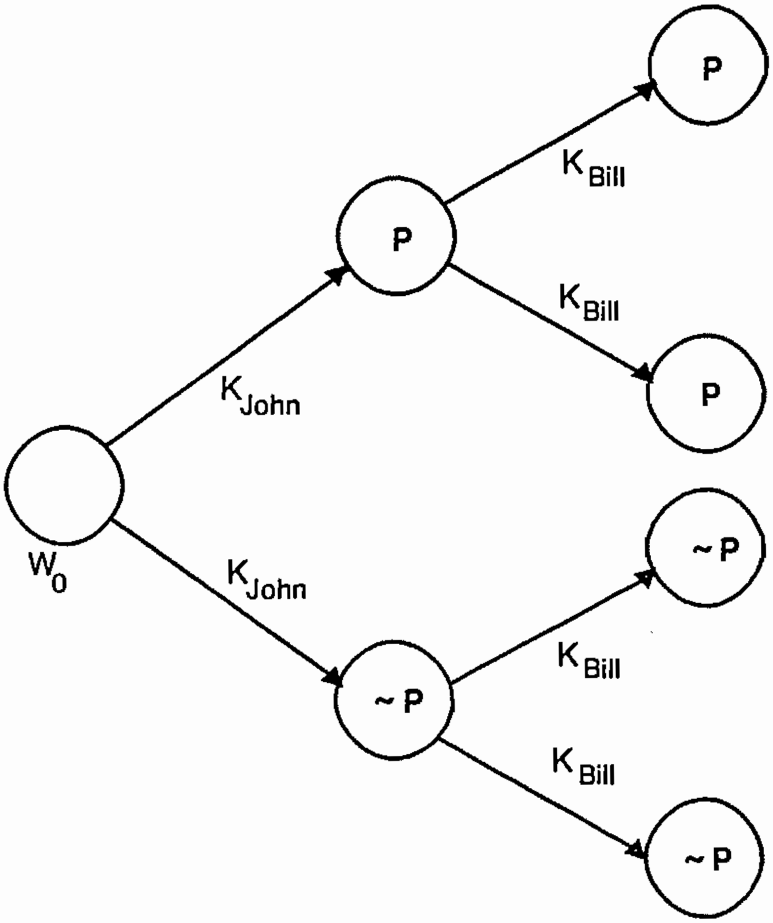

Figure 3.3

John knows that Bill knows whether *P*, but John does not know whether *P*.

When one moves beyond a purely propositional object-language one is faced with the fact that object-language terms can have different denotations in different possible worlds. For example, the term President(Usa, Year(1981)) can denote Jimmy Carter or Ronald Reagan, depending on which possible world one is talking about. One can then assert that "John knows that the President of the United States likes jelly beans," without making any claims that John knows who the President is. In other words it is possible for John to agree with the above statement but answer "I don't know" to the question "Does Ronald Reagan like jelly beans?"

The effect of having an intensional object-language is that one must reason explicitly about the denotation of an object-language term in the meta-language for each term that can have multiple denotations. There are some object-language terms, called rigid designators, that have the same denotation in every possible world. These terms are treated specially by the system, and play an important role in reasoning about whether an agent knows who or what something is. The details of this process are covered in the following section.

Since one must reason about the denotation of terms, a function D is introduced that maps an object-language term and a possible world into the denotation of that term in the given world. Thus, we might assert

$$D(W_0, \text{President}(\text{Usa}, \text{Year}(1981))) = D(W_0, \text{Governor}(\text{California}, \text{Year}(1968))).$$

A meta-language axiom schema is required to express the fact that two object-language terms are equal with respect to a possible world if and only if their denotations are the same in that world:

$$\forall w\, T(w, EQ(X, Y)) \equiv [D(w, X) = D(w, Y)]. \qquad (EQ1)$$

The introduction of quantifiers into the object language poses a few minor problems. These arise through the introduction of an object-language variable into a term that could have different values in different possible worlds. In an extensional object-language, any term that denotes the individual would suffice. However, since the object-language is intensional and the terms can have different denotations, we have to take into account whether the term that we substitute for the quantified variable will be evaluated with respect to a different possible world where it could denote a different individual.

This difficulty can be circumvented by always substituting a term that has the same denotation in all possible worlds, or in other words, a rigid designator. We introduce a function, @, that maps a meta-language term into an object-language rigid designator that has the same denotation as does the meta-language term in all possible worlds. Thus, the translations of the object-language existential and universal quantifiers into the meta language are done according to the following axiom schemata ($Q1$) and ($Q2$). $P[@(x)/?x]$ means that $@(x)$ is substituted for $?x$ in the term P wherever it occurs.

$$\forall w[T(w, \exists ?x\, P) \equiv \exists x\, T(w, P[@(x)/?x])]. \qquad (Q1)$$

and

$$\forall w\, [T(w, \forall ?x\, P) \equiv \forall x\, T(w, P[@(x)/?x])]. \qquad (Q2)$$

In the axiom schemata ($Q1$) and ($Q2$), since the @ function constructs rigid designators, the following axiom always holds:

$$\forall w, x\, D(w, @(x)) = x. \qquad (Q3)$$

3. Knowing Who or What Something Is

Knowing who or what something is is of primary importance in planning that involves actions that another agent is expected to carry out since the planning agent must decide whether the other agent's knowledge is sufficient to allow the formulation and execution of the plan. For example, for an agent to manipulate a piece of equipment, he must know what the piece of equipment is, what the tools are that he is to use, and where they are located.

An agent knows what an object-language term is if it denotes the same individual in all possible worlds compatible with the agent's knowledge. Stated in the logic, this statement is equivalent to the schema:

$$\begin{aligned}\forall w_1\, T(w_1, \mathbf{KnowsWhatIs}(A, X)) \equiv \forall w_2\, K(A, w_1, w_2) \supset \\ D(w_2, X) = D(w_1, X).\end{aligned} \qquad (K4)$$

One can take a similar approach to representing that someone knows which individual satisfies a certain property or set of properties. For example, to say that John knows who murdered Smith is equivalent to saying that there is some individual in the real world about whom John knows that he murdered Smith. In object-language notation this is expressed as

$$\mathbf{True}(\exists ?x\, \mathbf{Know}(\mathrm{John}, \mathrm{Murdered}(?x, \mathrm{Smith}))).$$

This example demonstrates why rigid designators are important to knowing who or what something is. One could imagine non-rigid substitutions for $?x$ in the above example that would make the statement trivial; for example, define a function MurdererOf(x) with its obvious meaning, and substitute MurdererOf(Smith) for x. If the existential quantifier in the above example is translated into the meta-language acording to rule ($Q1$), then only a rigid designator or rigid function (a function that maps rigid designators into rigid designators) can be substituted for $?x$, and non-rigid substitutions like MurdererOf(Smith) are ruled out.

4. Representing the Relationship between Knowledge and Action

Moore [74] has proposed an elegant means of formalizing the relationship between knowledge and action that has been adopted as the basis of the language-planning formalism. His idea is to use possible worlds to represent the state of the world resulting from the performance of an action. Thus, in addition to the role possible worlds play in describing the semantics of the intensional operators, they also play a role similar to that of situations in a situation calculus [62], [50]. One can then define a meta-language predicate $R(a, w_1, w_2)$ that is true if and only if w_2 is the world resulting from the performance of action a in world w_1, which gives us a way of stating how different possible worlds are related by the performance of actions.

One of the most important problems that arises in attempting to axiomatize actions of any kind is the *frame problem.* The frame problem is the problem of specifying for each action precisely what aspects of the world are changed and what remain the same after the performance of the action. Since most actions have a very localized effect on the state of the world, it would be ideal to have a convenient way to formally state the few things that do change and then say "everything else remains the same." Saying that "everything else remains the same" is difficult, since it seems as though one has either to have an extremely large number of axioms, or one must quantify over predicates. Moore adapted Kowalski's approach to stating frame axioms [50] to the possible worlds formalism. The key idea is to translate object-language predicates into meta-language functions that map individuals into intensional objects. One can then quantify over these intensional objects in stating that they either do or do not hold in a given possible world. It becomes possible to have the effect of quantifying over predicates in a first-order theory.

The following schema for the translation of object-language predicates into the meta-language will be adopted:

$$\forall w, x_1, \ldots, x_n\, T(w, P(x_1, \ldots, x_n)) \equiv H(w, :P(D(w, x_1), \ldots, D(w, x_n))). \qquad (T1)$$

$:P$ is a meta-language function that maps an individual into an intensional object that may or may not hold in a possible world.* The difference between H and T is that $T(W, P(A))$ means that the object-language formula $P(A)$ is true in the world W, while $H(W, :P(:A))$ means that the individual $:A$ has the property $:P$ in W. The best way to understand the role of the T and H predicates is by an analogy drawn by Moore [74] to the difference between **Eval** and **Apply** in LISP. When a function is **Apply**d, its arguments have already been evaluated with respect to the relevant environments. The H predicate is like **Apply**, and T is more like **Eval**.

Object-language functions and constants are treated analogously. An object-language function translates into an intensional object, like the intensional objects corresponding to predicates, which determines a different individual in each possible world. A function V is defined that maps a possible world and one of these intensional objects into the corresponding individual. Thus, the analogous axiom schema for the translation of object-language functions into the meta-language is

$$\forall w, x_1, \ldots, x_n\, D(w, F(x_1, \ldots, x_n)) = V(w, :F(D(w, x_1), \ldots, D(w, x_n))). \qquad (T2)$$

We now have a formal tool for stating frame axioms. The statement that "everything true in w_1 is true in w_2" can be expressed as

$$\forall p\, H(w_1, p) \supset H(w_2, p)$$

and the statement that "all functions and constants have the same value in w_1 and w_2" is

$$\forall c\, V(w_1, c) = V(w_2, c).$$

---

* The correspondence between functions in the meta-language and predicates in the object-language can be chosen arbitrarily — all that is needed is some simple way of knowing what predicate corresponds to what function. For this purpose, Moore adopted the ':' notation.

One may ask why it is not possible to quantify directly over object-language terms in stating frame axioms, since in the meta-language, one can talk about terms in the object-language. The problem is that it then becomes difficult to deal with more complicated assertions involving quantifying-in. For example, suppose one wanted to say that after an action mapping W_1 into W_2, P is true of everything in W_2 that it was true for in W_1. This would lead us to state a frame axiom such as

$$\forall x\, T(W_1, P(@(x))) \supset T(W_2, P(@(x))).$$

If we wanted to prove $T(W_2, P(A))$, it would be impossible to use the above axiom because A does not unify with $@(x)$. What is needed is some way of reasoning about the denotation of all the terms that comprise the object-language expression. By using the translation rules $(T1)$ and $(T2)$, we can use rule $(Q3)$ to reason about the denotation of $@(x)$. The frame axiom becomes

$$\forall x\, H(W_1, {:}P(x)) \supset H(W_2, {:}P(x))$$

, and the goal to be proved is $H(W_2, {:}P({:}A))$.

With the basic tools for stating frame axioms available, we can now describe how the performance of an action affects the knowledge of various agents. Moore stated axioms for describing the effects of an action on the agent performing the act; however, for language planning, several additional situations must be considered. When planning language, a speaker is always dealing with at least one other agent. If one agent performs an action of which the other agent is unaware, the agent performing the action must be able to represent the fact that the ignorant agent still believes what he believed before the action took place. Also, two agents may be mutually aware of an action, although only one of them is actually performing it. In such a case, one wants to state how the action affects mutual knowledge. Similarly, one agent, A_1, may perform an action that is observed by a second agent, A_2, without A_1's knowing that A_2 is observing and can see what is going on. Finally, there are actions such as speech acts that always involve at least two agents, where both of the agents are mutually aware of the performance of the action. In this section, I will describe the fundamental case of an action affecting the knowledge of an agent. The effect of actions on mutual knowledge will be discussed in Section 5 (of this chapter) on representing mutual knowledge. The axiomatization of multiagent speech acts is described in Chapter V.

Adequately describing the effects of an action on an agent's knowledge requires describing a relationship between two sets of possible worlds, namely the set of possible worlds compatible with his knowledge before performing the action, and the set of possible worlds compatible with his knowledge after performing the action. If an agent knows about an action in the sense of knowing all of its preconditions and effects, this relationship can be stated by saying that if w_1 and w_2 are related by agent A performing action E, then the worlds compatible with what A knows in w_2 are exactly those worlds that are the result of E happening in some world that is compatible with what A knows in w_1. This relationship, which is expressed in Figure 3.4, tells us exactly how what A knows after E happens depends on what A knows before E happens.

Figure 3.4 expresses that what is possible according to A's knowledge after performing an action is always the result of performing the action in some world that was possible according to his knowledge before performing the action.

Notice that in Figure 3.4, it is not the case that there is a world compatible with A's knowledge in w_2 for *every* world compatible with his knowledge in w_1. The

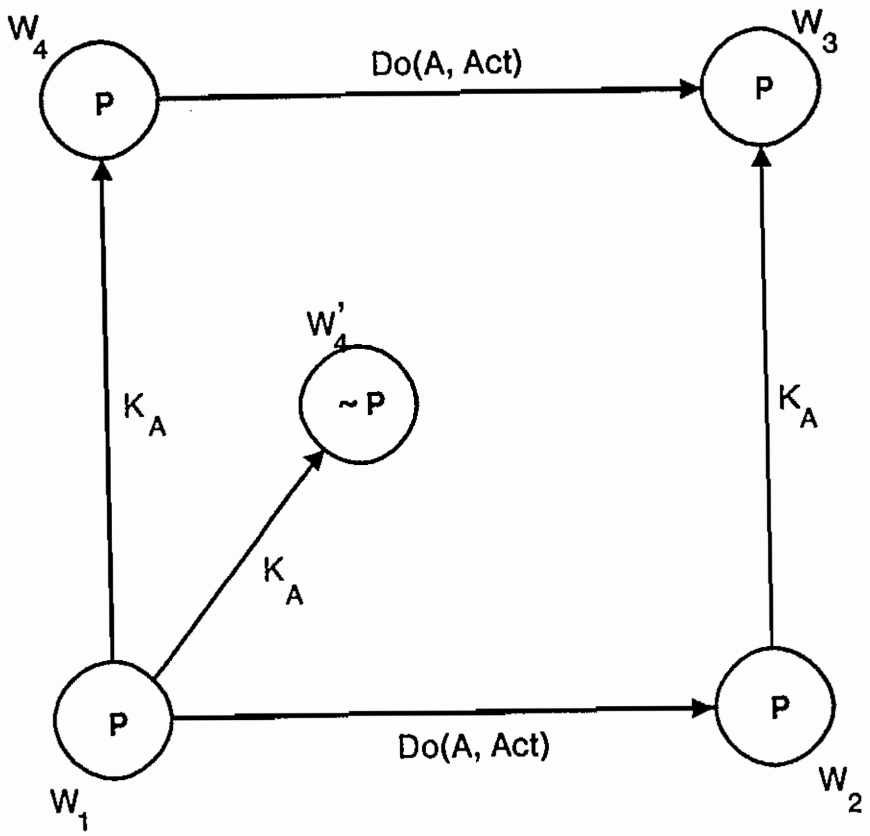

Figure 3.4

The effect of performing an action on the knowledge of the agent

reason for this is that it is possible for actions to ***produce knowledge*** by ***restricting*** the possible worlds that are compatible with an agent's knowledge after performance of the action. In world w_1, the agent does not know whether P is true, since both P and $\sim P$ are true in possible worlds compatible with his knowledge. However, after performing a knowledge-producing action, only worlds in which P is true are possible as far as he knows in w_2. In other words, performing the action has 'informed' the agent that P is true.

The principles involved here can best be illustrated by means of a simple example. Suppose we wish to axiomatize the action of removing one part from another in a disassembly operation — Do(A, Remove(x, y)). The preconditions for the action are that in the initial state, x must be attached to y, and A must be at the location of y. In the resulting state, x is no longer attached to y. Everything else stays the same as in the initial state.

The preconditions are expressed by a set of assertions about what must have been true in the initial state when it is asserted that an action is performed. Thus, the preconditions can be stated in the following axiom:

$$\begin{aligned}&\forall A, w_1, w_2, x, y\, R(\text{:Do}(A, \text{:Remove}(x, y)), w_1, w_2) \supset \\ &\quad H(w_1, \text{:Attached}(x, y)) \wedge \left[V(w_1, \text{:Location}(A)) = V(w_1, \text{:Location}(y))\right]\end{aligned} \tag{R1}$$

Notice that since axiom ($R1$) quantifies over all w_1, it is tantamount to asserting that the preconditions of removing are ***universally known***, since they hold in *all* possible worlds, including the worlds compatible with any agent's knowledge.

Next, we need an axiom that describes the effects of performing the action when the preconditions are satisfied. Such an axiom would look like ($R2$):

$$\begin{aligned}&\forall A, x, y, w_1, w_2\, R(\text{:Do}(A, \text{:Remove}(x, y)), w_1, w_2) \supset \\ &\qquad \forall P \left[((P = \text{:Attached}(x, y)) \supset \sim H(w_2, P)) \wedge \right. \\ &\qquad\quad \left. ((P \neq \text{:Attached}(x, y)) \supset (H(w_1, P) \equiv H(w_2, P)))\right] \wedge \\ &\qquad \forall z\, V(w_2, z) = V(w_1, z)\end{aligned} \tag{R2}$$

This axiom says three things: (1) The relationship of x being attached to y no longer holds in the world resulting from removing x from y, (2) Every other relationship remains unchanged from the original state, and (3) The values of all constants and functions are unaffected by the action.

The final required axioms are ones that relate agents' knowledge to the performance of the action. This is accomplished by asserting that the relationship illustrated in Figure 3.4 holds for the agent performing the action (and possibly those agents aware of the performance of the action) and that the knowledge of other agents "stays the same".

$$\begin{aligned}\forall A, x, y, w_1, w_2\, R(\text{:Do}(A, \text{:Remove}(x, y)), w_1, w_2) \supset \forall w_3\, [K(A, w_2, w_3) \supset \\ \exists w_4\, K(A, w_1, w_4) \wedge R(\text{:Do}(A, \text{:Remove}(x, y)), w_4, w_3)]\end{aligned} \qquad (R3)$$

What axiom $(R3)$ says in essence is that when an agent performs the *remove* action, he knows that he did it. In other words, every world that is compatible with his knowledge after performing the action is the result of doing the action in some world compatible with his knowledge beforehand. Since we have assumed that the preconditions and effects of *remove* are universally known to all agents, it is possible to prove using axiom $(R3)$ that the agent must know that the prerequisites held before performing the action, and that he knows the changes brought about by executing the action and any of their logical consequences according to his knowledge.

The axiom

$$\begin{aligned}&\forall A, x, y, w_1, w_2\, R(\text{:Do}(A, \text{:Remove}(x, y)), w_1, w_2) \supset \\ &\qquad \forall B, P, w_3\, [(A \neq B) \wedge K(B, w_2, w_3) \supset \exists w_4\, K(B, w_1, w_4) \wedge \\ &\qquad H(w_3, P) \equiv H(w_4, P) \wedge H(w_2, P)]\end{aligned} \qquad (R4)$$

expresses the fact that all agents other than the one performing the action are "ignorant" of the action, or in other words, after the performance of the action they know precisely what they knew before the event happened. The requirement that P holds in w_4 *and* w_2 is to express the fact that if an action that agent A performs unknown to another agent B changes some state of the world that A knew to be the case originally, then in the resulting state, B no longer *knows* it to be the case (although he may still *believe* that P holds. To correctly handle belief is by no means a trivial problem, and will not be considered at this time.)

5. Representing Mutual Knowledge

Chapter V outlines the necessity for reasoning about mutual knowledge in a language planning system. A and B are defined to mutually know P if A knows P, B knows P, A knows that B knows P, B knows that A knows P, A knows that B knows that A knows P, and so on to an arbitrary depth of each agent knowing about the other agent's knowledge. The primary problem presented by representing mutual knowledge is formulating a finite representation of an infinite number of facts. Since one cannot possibly store an infinite number of assertions, one must be able to arrive at some axiom or set of axioms that will allow the derivation of the knowledge about knowledge relationships to any arbitrary depth.

Cohen [15], [16] proposed a solution to this problem in which sets of assertions about what an agent believes are placed in possibly overlapping spaces in a partitioned semantic network. The set of assertions about a speaker's beliefs are placed on a space labeled SB. The assertions about the speaker's beliefs about the hearer's beliefs are placed on a space SBHB, nested inside SB. Mutual belief was represented by a circular link from SBHB to SB, which Cohen's deduction system interpreted as meaning that the hearer's beliefs were identical to the speaker's own beliefs. The derivation of the mutual belief assertions could be carried out to an arbitrary depth by chasing the circular pointers around.

Although a scheme similar to Cohen's might work, since there are independent justifications for choosing the possible-worlds semantics approach, we need a means of representing mutual knowledge that fits well within the possible-worlds framework. A special case of mutual knowledge has already been mentioned in the previous section, namely the case in which one wishes to represent that all agents mutually know a certain fact. This can sometimes be accomplished by asserting that the fact is necessarily true. A consequence of necessary truths is that they are true in every possible world compatible with any arbitrary agent's knowledge, and so therefore are mutually known by every agent. The necessary truth approach is most useful in stating mutual knowledge about things that are not likely to change over time, for example, the definition of actions mentioned earlier.

However, this means of talking about mutual knowledge as necessary truth will not work in all cases. Some things that one would want to assert to be universally known are, in fact, not necessarily true in most reasonable models of the world. For example, one may want to assert that it is universally known that the White House is white, but it is not necessarily white, since it is logically possible for some agent to paint it pink. This approach also fails when one wants to consider the common case of three agents A, B and C, where A and B mutually know P, but C does not know P.

An approach to representing mutual knowledge that is consistent with the approach outlined so far is a variation of Sato's "any fool" approach [63]. Sato axiomatized universal knowledge in solving the *Three Wise Men* problem by hypothesizing an individual called "any fool" and asserting that universal knowledge consists of those facts that "any fool knows." The ability to deal with some types of universal knowledge as necessary truth eliminates much of the need for any individual exactly like "any fool". However, a good solution to the mutual knowledge problem can be found by talking about hypothetical agents that play the role of an "any fool" with respect to sets of two or more agents.

The hypothetical "any fool" individual will be replaced by a function that constructs hypothetical individuals from a list of agents. In this example, I will consider only the case of describing the mutual knowledge of a set of two individuals, A and B. The function that constructs hypothetical agents is called the *Kernel* function, since it is intended to represent the kernel of knowledge that is shared mutually by A and B. The facts that are mutually known by x and y are precisely those facts that are known by the kernel of x and y. The function Kernel(x, y) maps two individuals onto their Kernel. Since the argument list of Kernel is unordered, the following axiom is also needed:

$$\forall x, y \ \text{Kernel}(x, y) = \ \text{Kernel}(y, x). \qquad (MK1)$$

What is needed now is a possible-worlds interpretation of the knowledge of Kernel(x, y). The interpretation that immediately suggests itself is to say that the set of possible worlds compatible with an agent x is a subset of the possible worlds compatible with the kernel of x and any other agent. This gives us axiom $(MK2)$:

$$\forall x, w_1, w_2 \, K(x, w_1, w_2) \supset \forall y \, K(\ \text{Kernel}(x, y), w_1, w_2) \qquad (MK2)$$

. It should be noted that saying that the worlds compatible with the kernel are a *superset* of the worlds compatible with the agent means that the kernel's knowledge is a *subset* of the agent's knowledge, since the more restrictions are placed on the worlds compatible with an agent's knowledge, the more the agent knows. The two axioms $(MK1)$ and $(MK2)$ are all that is needed to extend the formalism to handle mutual knowledge between sets of two agents. Figure 3.5 illustrates the relationship between possible worlds compatible with the mutual knowledge of two agents.

The dashed lines in Figure 3.5 relate the worlds compatible with the knowledge

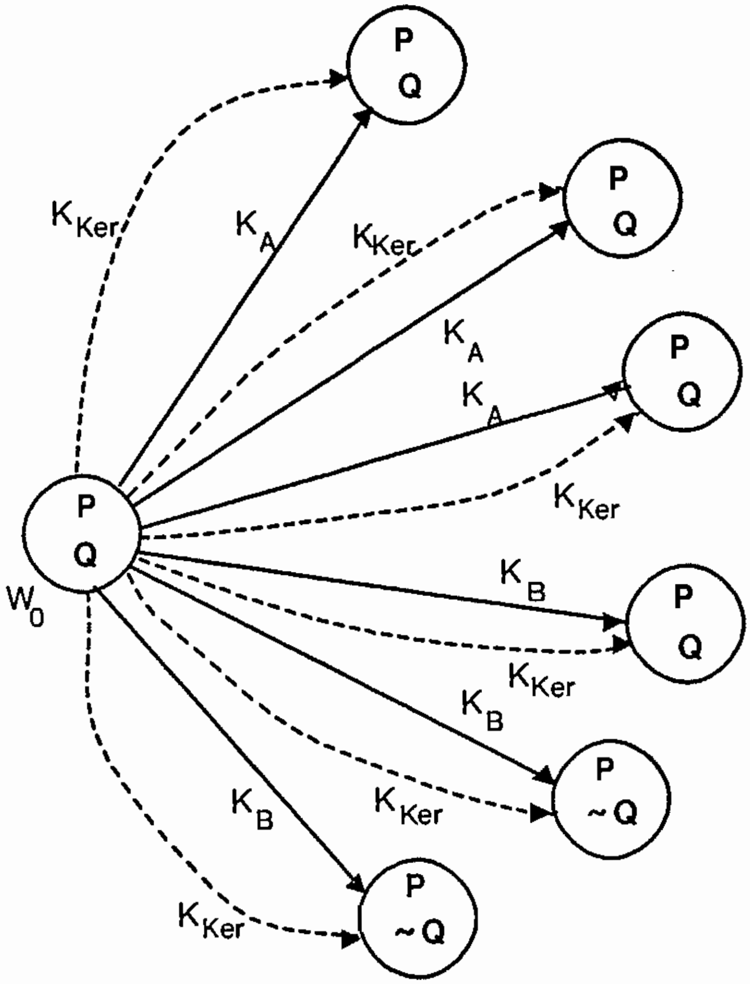

Figure 3.5

A and *B* mutually know *P*, *A* knows *Q*, *B* does not know whether *Q*

of the kernel. The diagram shows how different agents can know different things with the kernel representing the shared knowledge.

Additional axioms must be included with the axioms that describe actions to state the effects of the actions on mutual knowledge. This can be accomplished by an intensional operator that states that an agent is *aware* of an action. For example,

$$\begin{aligned}\forall w_1 T(w_1, \mathbf{Aware}(A, \mathrm{Do}(B, \mathrm{Act}))) \equiv & \forall w_2\, R(\mathrm{:Do}(:B, \mathrm{:Act}), w_1, w_2) \supset \\ & \forall w_3 \left[K(:A, w_2, w_3) \supset \right. \\ & \left. \exists w_4\, K(:A, w_1, w_4) \wedge R(\mathrm{:Do}(:B, \mathrm{:Act}), w_4, w_3)\right]\end{aligned}$$

This axiom says that the effects on knowledge of an agent that is aware of another agent's action is the same as the effect on the agent performing the action as described in axiom ($R3$) — the agent knows that the action has taken place. Axiom ($R3$) says that an agent is aware of his own actions, and this axiom generalizes this to other agents as well. If one asserts awareness of the kernel of the two agents, then they are *mutually aware* of the action, and they both mutually know that the action has taken place.

6. Reasoning about Wanting and Intention

Reasoning about what an agent wants is a very difficult problem for which only a limited solution is presented here. A representation system is proposed that allows one to represent the fact that an agent may have wants that are inconsistent with each other as long as his sets of simultaneous wants are logically consistent. An agent can also want states of affairs that are unachievable from the current state of the world.

There are some difficult philosophical problems with reasoning about wanting that do not seem to have any obvious solution. One problem is that of necessary truths — statements that are true in every possible world. Although it is certainly futile, and it may be irrational for an agent to want the negation of a necessary truth, it is certainly possible for rational agents to not care whether a particular necessary truth holds or not. Any representation that uses a possible-worlds semantics will suffer from the inability to represent the perfectly reasonable statement "*John doesn't care whether Fermat's Last Theorem is true,*" assuming that if Fermat's theorem is true, then it is necessarily true, and if it is false, then it is necessarily false. Any possible-worlds representation of "doesn't care" done along the lines of "doesn't know" entails stating that a proposition is true in some possible worlds compatible with an agent's wants and not true in others.

Another difficult problem is describing how an agent's wants are affected by actions. The effects of knowledge on an agent who performs an action can be described strictly as a function of the action itself; it does not matter what the prior intentions of the agent are to describe what happens to an agent's knowledge when he performs an action. In contrast, the effects of performing an action on an agent's wants is much more difficult to describe. An action may produce knowledge that in turn affects what the agent wants. Even more difficult to describe is that the actions an agent wants depend on how the actions fit into his overall plan. For example, if an agent has a plan of doing action A_1 followed by A_2, then in the initial state of the world, it is reasonable to say that the agent wants to do A_1. After the agent has done A_1, he no longer wants to do A_1, but now he wants to do A_2. The change in the agent's wants is not directly caused by any property of the actions he performed, but rather caused by a change in the state of the agent's plan as a result of executing part of it. Therefore, a fully adequate treatment of wanting and intention must entail an adequate representation in the logic of what it means for an agent to have a plan, and to execute part of a plan. The effects on an agent's wants would be described as the effects of a "meta-action" of executing a step in a plan. A full discussion of the problems involved and possible solutions is beyond the scope of this work. Some work on meta-planning (e.g., [103]) may be relevant to the problem, since in such a formalization, planning can be viewed as an action that has its own effects, possibly changing the wants of the planning agent.

There is also a spectrum of distinctions that can be drawn among an agent's wants. Some wants may be desires that the agent knows to be unrealizable (e.g., "I wish my father was still alive"), some may be achievable, but the agent does not know of any plan for achieving them (e.g., "I want to have a million dollars"), others may be wants that do not get translated into intentions because of competing contradictory desires (e.g., "I want to have a candy bar, but I want to watch my weight"), and finally, there are wants that are realized as intentions of actually performing some action (e.g., "I want to drink a Coke, so I shall walk down the hall to the machine and insert my money").

The proposal presented here is a first cut at enabling a system to deal adequately with a limited domain and needs to be considerably expanded. I will not attempt to deal with degrees of wanting, nor attempt to reason about conflicts between competing wants. The mechanism presented here can reason about particular wants an agent has, and draw simple conclusions from them. This will be sufficient for our purposes for the time being.

The reason for having at least a simple means of reasoning about an agent's wants in a language planning system arises from the necessity of forming multiple agent plans. Since communication in a task-oriented domain arises from the need of two or more agents to form a shared plan for accomplishing a task, there is a need for one agent to be able to talk about what another agent wants, and intends to do. Whenever an agent is making a plan that involves any agent other than itself performing an action, the agent must somehow know that the other agent wants to do the action in question, otherwise he would have no assurance that the plan would work.

The implicit assumptions made by the system that will enable it to function without the complex machinery of a complete ability to reason about wanting and intention are (1) that all actions are mutually known to all agents, and (2) that plans may be shared by two or more agents.

The first assumption means that the preconditions and effects of all actions are known to all agents, and that all agents know how to do all actions. This assumption is not as restrictive as it sounds at first. For example, this assumption means that all agents know what it means to remove *part1* from *part2* in the sense that they know that the action entails locating all the fasteners that connect *part1* to *part2*, locating the tool appropriate for each fastener, and undoing the connection. Under this assumption, there are still many points at which a planner may be blocked by a lack of knowledge. For example, the agent may not know what the fasteners are or where they are located, he may not know what the right tool is for an unfastening operation, and he might not know where the tool is located.

The simplification achieved by this assumption is that the planner is entitled to assume that all other agents can expand their goals into plans provided that they have the right knowledge about the state of the world. Examination of actual expert-apprentice task-oriented dialogs collected as part of the research on the TDUS system [20], [83] reveals that this assumption is usually satisfied in practice. The apprentice always knows in general what it means to remove something — he doesn't know in all cases how the removal operation is to be instantiated in a particular instance.

If actions can be assumed to be mutually known, the planner can assume parts of its plans can be shared. If the planner can show that an agent wants to do a high-level action, then all the actions constituting the expansion of the high-level action can be assumed to be a shared plan between the planner and the other agent. The planner can assume that each agent can make the same plans that it can, using intensional descriptions of objects in the domain. Thus, the problem of reasoning at each step whether an agent wants to do the next action can be eliminated if it is assumed that the plan is shared. The reasoning about the agent's wants need only be done at the top level to establish that he wants to achieve the high-level goal.

Representing what an agent wants is similar to representing what he knows. The fact that an agent intends to do a particular action is represented by an object-language predicate WANTS-TO-DO(A, X), which means that agent A wants to do action X. Representing the fact that an agent wants a particular proposition to be true is more difficult because wanting must be represented as a sentential operator similar to **Know**. Thus, it is possible to talk about somebody wanting someone to know something, as well as knowing that somebody wants something.

A meta-language axiomitization of the possible-worlds semantics of the **Want** operator must be formulated. It will be adequate here to formalize only a very weak notion of wanting in a manner similar to **Know**. A relation W is defined on possible worlds such that world w_1 is related to w_2 if and only if w_2 is compatible with what agent A wants in world w_1. Since we would like agents to be able to have wants that are mutually contradictory, we partition the set of possible worlds compatible with an agent's wants into several sets of worlds, each of which represents one compatible set of the agent's wants. We can enumerate these partitions, and then add an additional argument to the W relation to indicate which partition we are talking about. Therefore, to represent the statement

$$\mathbf{Want}(A, P)$$

, we can say in the meta-language

$$\exists i \forall w_1\, W(A, i, W_0, w_1) \supset T(w_1, P) \tag{W1}$$

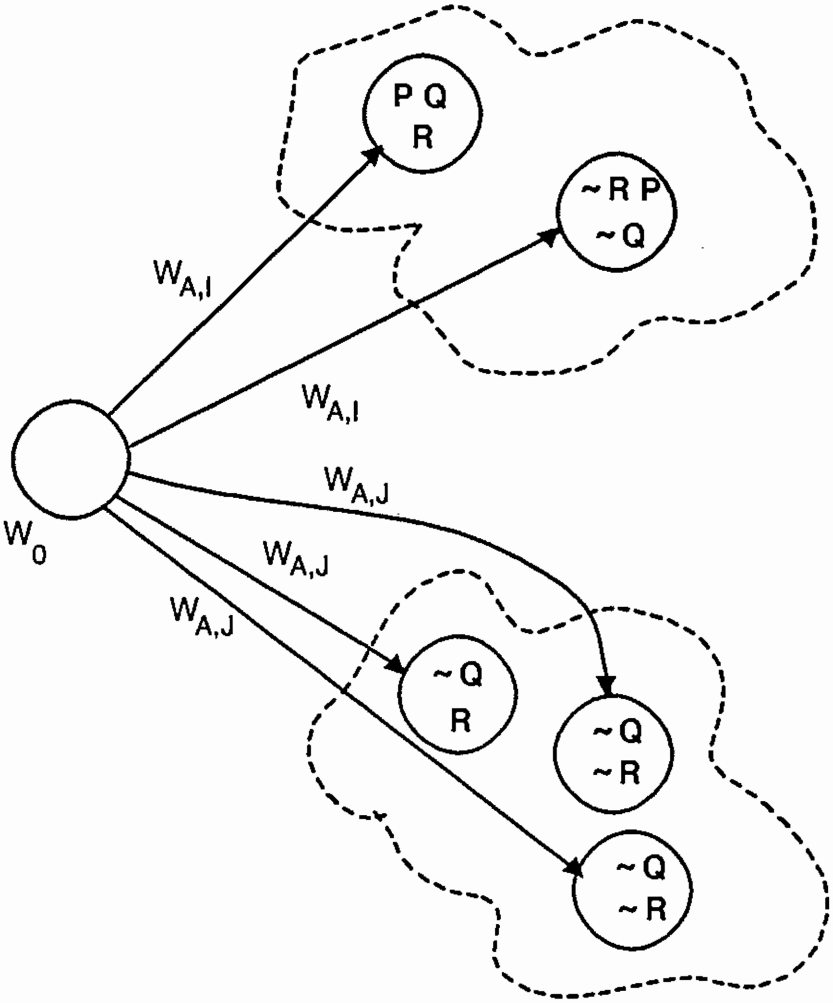

Figure 3.6

A wants *P*, *A* does not want *Q*, *A* doesn't care whether *R*

The relationship represented by the above axiom is described pictorially in Figure 3.6. *A* wants *P* because there is a want-set in which *P* is true for every possible world in that set. *A* does not want *Q* because there is no want-set for which *Q* is true in every possible world. In fact *A* wants $\sim Q$, since there is a want-set such that $\sim Q$ is true in every possible world. *A* doesn't care whether *R*, since it is not true that *A* wants *R* and it is not true that *A* wants $\sim R$ under the given partitioning. It should be noted that the partitions of worlds are not arbitrary, they are induced by *A*'s wants, and specific axioms are needed to describe what this partitioning is.

An agent wants P *and* Q if and only if P and Q are true for every possible world in the same want set. Thus it is possible to prove

$$\mathbf{Want}(A, P) \wedge \mathbf{Want}(A, \sim P)$$

, provided that each conjunct can be proved with respect to different want sets. It is never possible to prove $\mathbf{Want}(A, P \wedge \sim P)$.

One of the most common inferences the system makes about wanting is that if one agent is helpfully disposed toward another, and he knows that the other agent wants something, then he wants that for himself. This relationship makes one connection between knowledge and wanting, and is shown in Figure 3.7.

What Figure 3.7 says intuitively is that if A knows that some world is consistent with what B wants, then that world is also consistent with what A wants, with respect to some want-set. This is, of course, a very simplified version of what is actually the case. It is seldom true that a person will want everything that he knows another person wants. However, if the domain of discourse is restricted to a cooperative endeavor (e.g., the task in a task-oriented dialog), this assumption is adequate to produce reasonable behavior, because it is then reasonable to assume that the expert and the apprentice will cooperate whenever possible to complete the task. The only situation in a task-oriented dialog where this simple approach fails is when the apprentice forms an incorrect plan for carrying out the task, and then wants to achieve goals and perform actions that are not part of a correct plan. To avoid problems of reasoning about incorrect beliefs, we have made the additional simplifying assumption that an agent will make a correct plan if he can make any plan at all, thus the problem of inconsistent wants is avoided as well.

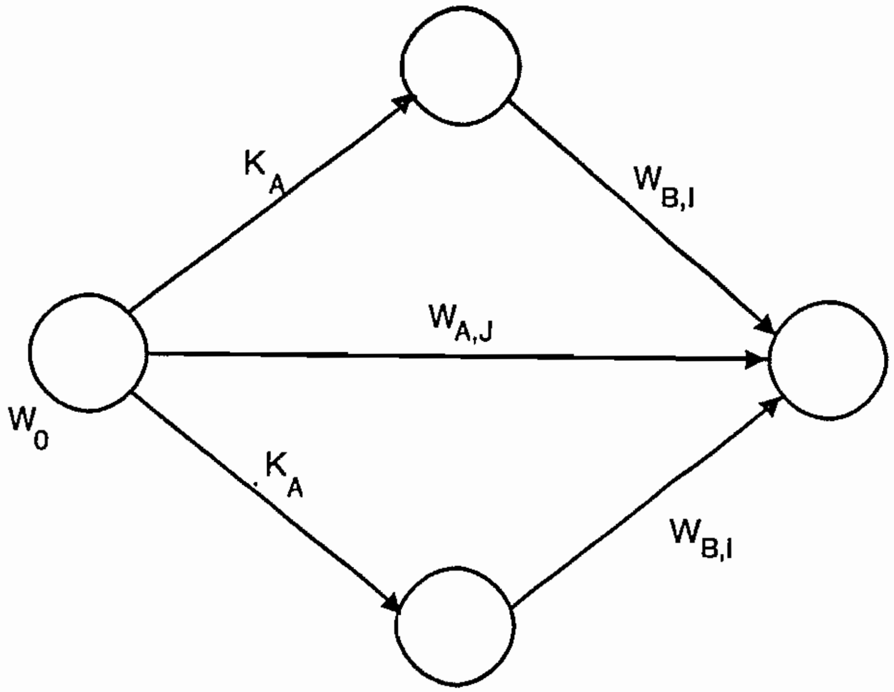

Figure 3.7

A wants what he knows that *B* wants

7. Conclusion

This chapter has developed a formalism that can serve as the basis for a language planning system. As with any formalism, it has both desirable features and some inherent limitations. The desirable features include the power to represent and enable one to reason about knowledge, for example, the ability to state that somebody knows the answer to a question without stating what the answer is, to talk about somebody knowing what something is, and knowing about an action.

The inherent limitations include the inability to conclude that a person does not believe a logical consequence of his knowledge, and the inability to express wanting with respect to necessary truths.

A number of simplifying assumptions have been pointed out in this chapter to avoid having to deal with very difficult problems that are related only tangentially to this research. It is important to realize that the difficulties these assumptions are intended to avoid are not *inherent* limitations of the formalism according to the best of my knowledge. For example, the representation presented here could possibly be extended to nonmonotonic reasoning along the lines of Doyle [22] to permit a reasonable treatment of belief and belief revision. More sophisticated axioms and deduction techniques could be applied to reasoning about wanting to draw conclusions about what an agent will do when faced with contradictory wants. Whether the formalism will actually be adequate to handle these more difficult problems, or whether some other scheme will be more fruitful, is an interesting empirical question to be settled by future research. However, for the time being, there are some pressing problems in reasoning about natural language for which the approach outlined here provides a reasonable place to start toward a solution, and enables the system to reason about utterances and their role in a dialog in a manner that has not been attempted by a language generation system to date.

IV

PLANNING TO AFFECT AN AGENT'S MENTAL STATE

0. Introduction

This chapter deals with the design and implementation of a planning system called KAMP (an acronym for **K**nowledge **A**nd **M**odalities **P**lanner) that is capable of planning to influence another agent's knowledge and wants. The motivation for the development of such a planning system is the production of natural-language utterances. However a planner with such capabilities is useful in any domain in which information-gathering actions play an important role, even though the domain does not necessarily involve planning speech acts or coordinating actions among multiple agents.

One could imagine, for example, a police crime laboratory where officers bring substances found at the scene of a crime for analysis. The system's goal is to know what the unknown substance is. The planner would know of certain laboratory operations that agents would be capable of performing, and these actions would produce knowledge about what the substance is or is not. A plan would consist of a sequence of such information-gathering actions, and the effect of executing the entire plan would be that the agent performing the actions knows the identity of the mystery substance. Since the primary motivation for KAMP is a linguistic one, most of the examples will be taken from language planning, but the reader should note that the mechanisms proposed are general and appear to have interesting applications in other areas as well.

1. The Problems of Planning to Affect Mental States

Most planning systems that have been developed to date have been designed to cause discrete state changes among a set of discrete objects. Many of the planning systems were applied to a "blocks world" in which the task was to move toy blocks of different shapes and colors into different configurations. Even planners operating in "real world" domains fall into this category to the extent that their domains are formalized as discrete state changes among discrete objects, making the domain isomorphic to some blocks world.

Although it may be tempting to think of blocks-world planning problems as trivial, there are many problems in planning in such domains that have yet to be settled [100]. Even so, it is becoming increasingly necessary to move beyond the restrictive assumptions of blocks-world domains. One assumption that can be weakened is the assumption of discrete state changes. Some planning work has proceeded in this direction to allow the description of continuous processes and simultaneous events (Hendrix [41], McDermott [66]). For the work described in this thesis, it will be adequate to retain the simplifying assumption of discrete state changes, but we will be forced to relax the assumptions about discrete objects. The planner's world will still be populated with discrete objects, but the planner will have to consider mental states as well, which have properties different from ordinary physical objects, and therefore require different planning techniques.

One approach to planning to affect mental states is to treat the mental states as discrete objects and manipulate them as such. This is usually done by assuming that intelligent agents have a "data base" of assertions of things they believe about the world. Planning operators that affect knowledge, such as informing, are formalized so that they insert or delete assertions from an agent's data base. A variation of this technique was used by Cohen [15] in his speech-act planner, and a similar, but more sophisticated, approach has been advocated by Konolige and Nilsson [48] for planning in a multiple agent environment.

Proposing a data base for representing an agent's beliefs encounters a number of well-known problems, discussed in detail by Moore [74]. The most serious objection is that in some versions of such a scheme it is difficult to talk about what an agent *does not* know (as opposed to what he knows not to be the case). Cohen proposed asserting $\sim \mathbf{Believe}(A, P)$ in a global data base, entirely separate from any agent's knowledge base. This approach may make the necessary representational distinctions, but it becomes very cumbersome to reason with the knowledge when a large number of such assertions must be made. The problem is particularly serious when one needs to combine facts from a particular data base with global facts to prove a single assertion. For example, from

$$\sim \mathbf{Know}(\mathrm{John}, Q) \wedge \mathbf{Know}(\mathrm{John}, P \supset Q)$$

where $P \supset Q$ is in John's model of the world (the "data base" for John), and $\sim \mathbf{Know}(\mathrm{John}, Q)$ is asserted in the global data base, it should be possible to conclude $\sim \mathbf{Know}(\mathrm{John}, P)$. A good strategy for combining information from these multiple sources has yet to be demonstrated.

Konolige and Nilsson employ a meta-language and a reflection principle to encode knowledge of the form $\sim \mathbf{Believe}(A, P)$, following Weyhrauch [101]. Although their approach is more sophisticated and overcomes some of the objections to syntactic approaches based on consistency (see Montague [71]), it is still an open question as to whether this technique can be used efficiently to solve problems such as the one above.

The possible worlds semantics approach to representing knowledge, discussed in Chapter III, circumvents many of the difficulties inherent in the data-base approach. Unfortunately, planning within this formalism presents problems that have not been faced by planning systems designed to date.

In contrast with the data-base approach, the possible-worlds-semantics approach represents a mental state by a collection of possible worlds consistent with the state rather than by an explicit list of assertions, thereby implicitly representing the assertions that are true. Mental states are still "objects" in the sense that they are entities that can be manipulated by the performance of actions, but they do not exist in possible worlds the same way that physical objects do, which presents some problems for a planning system. Achieving that A knows that P requires making P true in every world that is compatible with A's knowledge. The form of this goal is quite different from the goals that previous planning systems have dealt with, which consisted of formulas with only existentially quantified variables. A goal involving an agent's knowledge must quantify over all the possible worlds compatible with an agent's knowledge — a potentially infinite set.

Another difficulty that arises with any formalism intended for this type of planning is the problem of agents being able to reason with their knowledge. If the planning agent A_1 knows that an agent A_2 knows that $P \supset Q$, and A_1 has the goal that A_2 knows Q, then it should be possible for A_1 to achieve his goal by bringing it about that A_2 knows P. The formalism proposed here for representing knowledge about actions requires this type of reasoning to be done quite frequently when reasoning about what an agent knows after an action is performed. In order for A_1 to know any particular effect on A_2's knowledge from A_2 performing an action requires reasoning like "A_2 knows he has just performed an action, I know that he knows what the effects of the action are, therefore I can conclude that he knows the change in the world brought about by any particular effect of performing the action."

The general problem of finding the right bit of knowledge that an agent needs to perform a deduction can be quite difficult. Without any heuristics to guide the search, it would have to proceed by a process like the following: A_1 has the goal of A_2 knowing Q, but for some reason it is impossible or undesireable for A_1 just to inform A_2 that Q. Perhaps informing A_2 that Q would require activating concepts for which A_1 has no description. In such situations, the planner must attempt to achieve Q by finding a subgoal P such that A_1 knows P but A_2 does not know P. A_1 can then plan to inform A_2 that P, which may be possible, whereas the first informing action was not.

The problem is that the number of subgoals like P that must be considered expands very rapidly. Allowing a planner to do completely general reasoning about what an agent needs to know forces it to search through an extremely large space with little to guide the search.

2. Planning within the Possible-Worlds Formalism

Fortunately, many of the situations in which an agent must plan to tell another agent something are more tightly constrained than the general case because they fall into categories in which good heuristics exist to guide the search for a solution. KAMP solves problems by employing a heuristic problem-solving method that is successful at finding a good plan with minimal effort most of the time while preserving the option to rely on brute-force search if heuristic methods fail.

Early problem-solving systems such as STRIPS had the advantage of having a simple indexing scheme that could tell what actions are used to achieve particular goals. This was combined with an assumption restricting the predicates used in goals to be those that actions were capable of affecting. For example, if there was a goal of the form On(A, B), STRIPS had only to search its index for some action that had an assertion on the add-list that unified with the goal. It was always obvious what actions were potentially useful from the description of the effects of the action.

Because of the way actions are axiomatized in the formalism we are adopting, it is impossible to assume that the predicates that describe an axiom's effects will always match the predicates that occur in goals, since quite frequently the only effect of an action will be the assertion of a restriction on a relation between possible worlds and an agent. One inference that must be made frequently is that if an agent knows what the effects of an action are and he knows that the action has been performed, then he knows what changes have come about in the world as a result of the performance of the action. Allowing this inference means that one does not need two redundant lists of effects for each action: the effects of the action on the world and the fact that the agent knows each of the effects. In addition to this benefit, the generality of the approach allows one to reason that if A_1 knows that A_2 performed an action, then A_1 knows what changes occured, even though the axiom never explicitly mentions anything about A_1's knowledge. The generality of this approach does more than simplify the axioms, it extends the system's power to reason about knowledge and action. It is therefore desirable to find some means of retaining the generality while making it possible to plan efficiently.

Many solutions to this problem are quite unappealing. One could do a blind forward search, trying all actions to see if the desired effect could be achieved.

However, planning is difficult enough without removing all the constraints from the search space. Another solution is to put up with redundancy and axiomatize the knowledge-effects of each action individually. The problem goes beyond mere redundancy, however. The effect of a knowledge-producing action like informing depends on what the hearer knows when the action is performed. The problem of specifying in advance *all* the possible consequences of an action seems to be more difficult than the original problem.

The solution adopted by KAMP is to have two descriptions of the actions available to the planner. One description is in the form of axioms relating possible worlds as described in Chapter III. The axioms describe the actions precisely and in rich detail. The other description is an *action summary*, which summarizes the preconditions and effects of actions in a STRIPS-like formalism (see Fikes and Nilsson [23]) involving preconditions, add and delete lists. The action summaries are used by the planner as a heuristic to guide the selection of actions that are likely to result in a good plan. They are not intended to be complete descriptions of all the consequences of performing the action. The axiomatization is used to reason whether the proposed plan is going to work. If the action summaries are well designed, the planner will propose correct plans most of the time, and the search required for finding a correct plan will be significantly reduced.

The search is facilitated by the simplifications introduced by the action summaries. For example, an implicit assumption in the action summaries is that all agents know what the effects of the actions are.* In some relatively rare instances this assumption may not hold, and any plan proposed that depends on that assumption will fail the verification step. The action summaries are used by a process that can be viewed as a "plausible move generator," proposing actions that are likely to succeed in achieving the goal.

As an example of how action summaries work, consider the example of an action summary for the INFORM action, the axiomatization of which is described in detail in Chapter III. The action that is being described is more precisely $\mathrm{Do}(A, \mathrm{Inform}(B, P))$, where A is the agent performing the action (i.e. the speaker), B is the hearer, and P is an object-language proposition that is the object of the INFORM. The axiomatization states that $\mathbf{Know}(A, P)$ and $\mathrm{Location}(A) = \mathrm{Location}(B)$ are prerequisites, that all agents know this, and the effect when the INFORM is successfully executed is that B and A mutually know that the INFORM has taken place. B can deduce from this knowledge that P is true, and therefore in the resulting state, B knows P. The action summary should provide a simple way of concluding that informing actions are usually a good strategy to try to get somebody to know something. The action summary would have $\mathbf{Know}(B, P)$ listed explicitly as a knowledge-state effect of the informing action, although the conclusion is only inferred from the axioms.

Prerequisites are also listed as part of the action summary, but there are a number of prerequisites called *universal preconditions* that are not listed explicitly because they apply to every action. There are few, if any, preconditions involving

---

* This assumption is really much less restrictive than it sounds. It means that an agent knows the nature of the *immediate* effects of his actions, not that he knows all logical consequences of the action. In other words, an agent could know the effects of removing part A from part B, but be ignorant of the fact that part C is attached to part A. In the resulting state, the fact that C is no longer attached to the assembly is a 'consequence' of his action that he does not know, assuming it cannot be directly observed.

the physical state of the world that can be said to apply to every action; however, there are some knowledge-state prerequisites that are both universally applicable and nontrivial. These knowledge preconditions can be summarized by the statement that an agent has to have an executable description of a procedure to do anything. This means that for each intensional description of any participant in an action, the agent must know to what that intensional description refers. For example, if an agent wants to perform an action like "PointAt(Murderer(Smith))," he must know what "PointAt" is, and what individual "Murderer(Smith)" denotes.

In Moore's original treatment of possible-worlds semantics, there was an intensional operator **Can** that was used to capture the notion of universal preconditions. The formula **True**(**Can**(Do(A, X), P)) means that P is true in the state of the world resulting from A doing X, and that all the necesary conditions on A's knowledge are satisfied. Since Moore was interested primarily in deducing that a given plan achieved a particular goal and not in finding the plan in the first place, it was possible to separate the universal preconditions in this manner. In planning, however, the universal preconditions are not really any different than other preconditions. Some may be satisfied in a particular state and others not, and plans must be developed to achieve the ones that are not. Requiring the planner to include the universal preconditions of each action planned captures the generality of Moore's approach while offering enough flexibility for planning.

The use of action summaries can simplify the process of searching for plans when a significant amount of deduction must be done to find out which action is applicable to achieve a particular goal. In reasoning about what is true in the states between the performance of actions, as when deciding whether or not preconditions are satisfied, the possible worlds axioms can be used directly. This approach allows one to have the descriptive power of the possible-worlds knowledge representation while preserving some of the efficiency advantages of simpler approaches.

3. Hierarchical Planning with KAMP

The planning system described is similar to planners like STRIPS in which all actions are described on the same level of detail, and finding a plan consists of finding a linear sequence of these actions that result in a state in which the goal is true. It has been frequently observed (e.g., Sacerdoti [86]) that searching such an unstructured space can be quite inefficient. A good heuristic for searching such a space is to first construct a high-level plan that ignores some of the effects of the actions, and on a second pass consider the more detailed effects and make minor adjustments to the overall plan to take the greater detail into account. Of course it is not necessarily true that the adjustments required will be minor. It is not difficult to construct pathological examples where the interaction of the effects of two actions requires complete revision of the entire plan. It merely seems to be a reasonable heuristic to apply to problems in many domains, and experience supports this conclusion.

The planning of linguistic actions is an example of a domain in which hierarchical planning is a good technique to use. There are at least two clearly defined levels of abstraction — that of deciding to perform an action, such as informing or requesting that will influence the mental state of another agent, and that of constructing an utterance that will realize high-level speech acts. The low-level process of constructing an utterance can benefit from the information in the high-level plan when deciding how to integrate multiple actions into the utterance and vice-versa, as described in Chapter VI. Hierarchical planning allows the division of the language production process into levels of abstraction while allowing the interaction between levels that is essential for language planning.

4. KAMP's Data Structures

KAMP is a hierarchical planner whose basic design is similar to Sacerdoti's NOAH planner [86]. The control strategy and data structures employed by the two systems are quite similar, although they differ in minor respects. The underlying representation and deduction systems upon which the two systems are based are radically different, and the problems caused by planning in a multiple agent environment also leads to some differences.

The data structure that KAMP uses to represent plans is called a *procedural network* [86]. The distinguishing feature of procedural networks is that they allow action sequencing information to be specified as minimally as possible. It is possible to represent plans as partially ordered sequences of actions, and a linear ordering of actions need be imposed only when sufficient information has been gathered so that one can avoid committing oneself to an incorrect linear ordering that will have to be discarded.

A procedural network can be thought of as a two-dimensional data structure. The horizontal dimension is a temporal one, which reflects the partial ordering among the actions. The vertical dimension is one of abstraction, where goals and abstract actions are refined into sequences of low-level executable actions. Figure 4.1 is an example of a simple procedural network.

Goals and actions are represented in the network as PLANSTEP nodes, shown as rectangular boxes in Figure 4.1. KAMP represents both goals and actions in the network. Goals can be thought of as very high-level actions, with vaguely

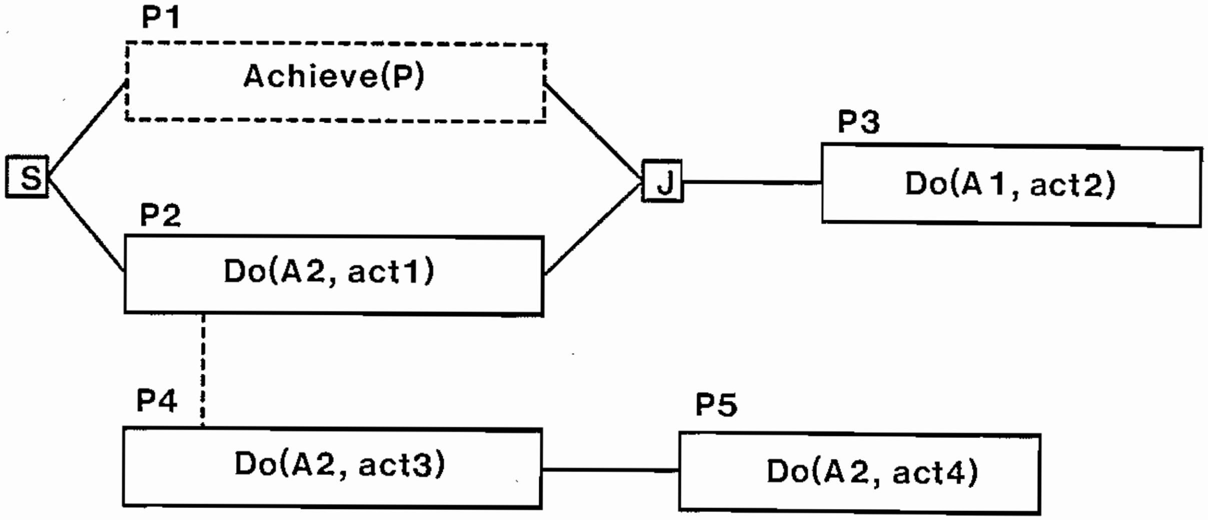

Figure 4.1

A Simple Procedural Network

specified conditions on what is true after the action is perfromed. The planner knows that the goal will be true in the resulting state, but it cannot yet reason about what has changed as a result of bringing it about. Node $P2$ in Figure 4.1 is a PLANSTEP for a high-level action, and $P4$ and $P5$ are low-level expansions of $P2$. *Phantoms* are goals that are already true in the current state of the world, so nothing has to be done to achieve them. They are represented in the diagrams by boxes consisting of dotted lines like $P1$ in Figure 4.1. Phantom goals are kept as part of the plan, because subsequent changes to the partial order of the actions may make it necessary to "undo" the effects of a previous action, and thus "unsatisfy" a phantom goal. Actions are represented by PLANSTEP nodes that contain a meta-language description of the action to be performed. It is possible for high-level actions to

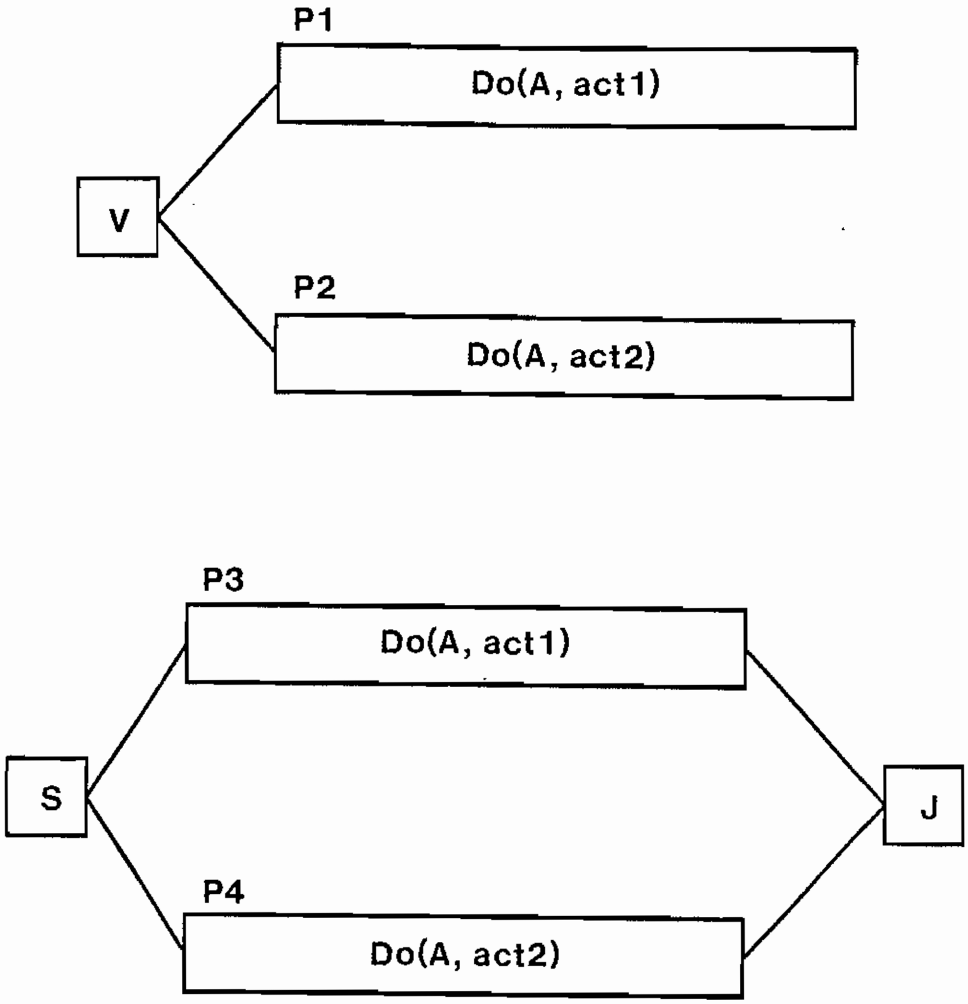

Figure 4.2

Choice and Split Nodes

be *subsumed*, which means that their principle effects are achieved through minor alterations in the low-level expansion of another action in the plan, rather than by direct expansion to the lower level. Speech acts are often subsumed, and this process is discussed in greater detail in Chapter VI.

There are two types of nodes in the plan, which represent alternatives between plan steps. *Choice* nodes split the plan into several parts depending on which one of several alternatives is selected. The goal could be achieved by executing *either* of the branches $P1$ or $P2$ in Figure 4.2. If the expansion of one of the branches of the choice fails, then it is pruned from the plan, and the other branches of the choice are expanded. The *split* nodes implement the partial ordering between plan steps. Each branch of a split must eventually be executed for the plan to succeed, but there is no commitment at that level to the order in which the branches are executed. In KAMP, splits are not intended to represent concurrent actions, and the planning formalism described here has difficulty with concurrent actions because of the use of possible worlds as discrete states brought about by the performance of single actions. The split expresses that there is no commitment to ordering the branches of the plan at some stage in the planning process. A linear ordering will eventually be chosen, arbitrarily if no better reason presents itself, but the decision will be postponed as long as possible.

It is also possible to describe nodes for iterated plan steps and conditional branches, but situations in which these constructs are necessary will not occur in any of the examples to be considered.

The connection between the planning data structure and the possible-worlds-semantics formalism is made by associating with each node of the plan a world that represents the actual state of affairs at each point. Whenever a fact has to be proved to hold in the situation resulting from the execution of a series of actions, it is proved using the world associated with the appropriate node in the procedural net as the current real world.

Figure 4.3 illustrates how worlds are associated with the expansion of a high-level

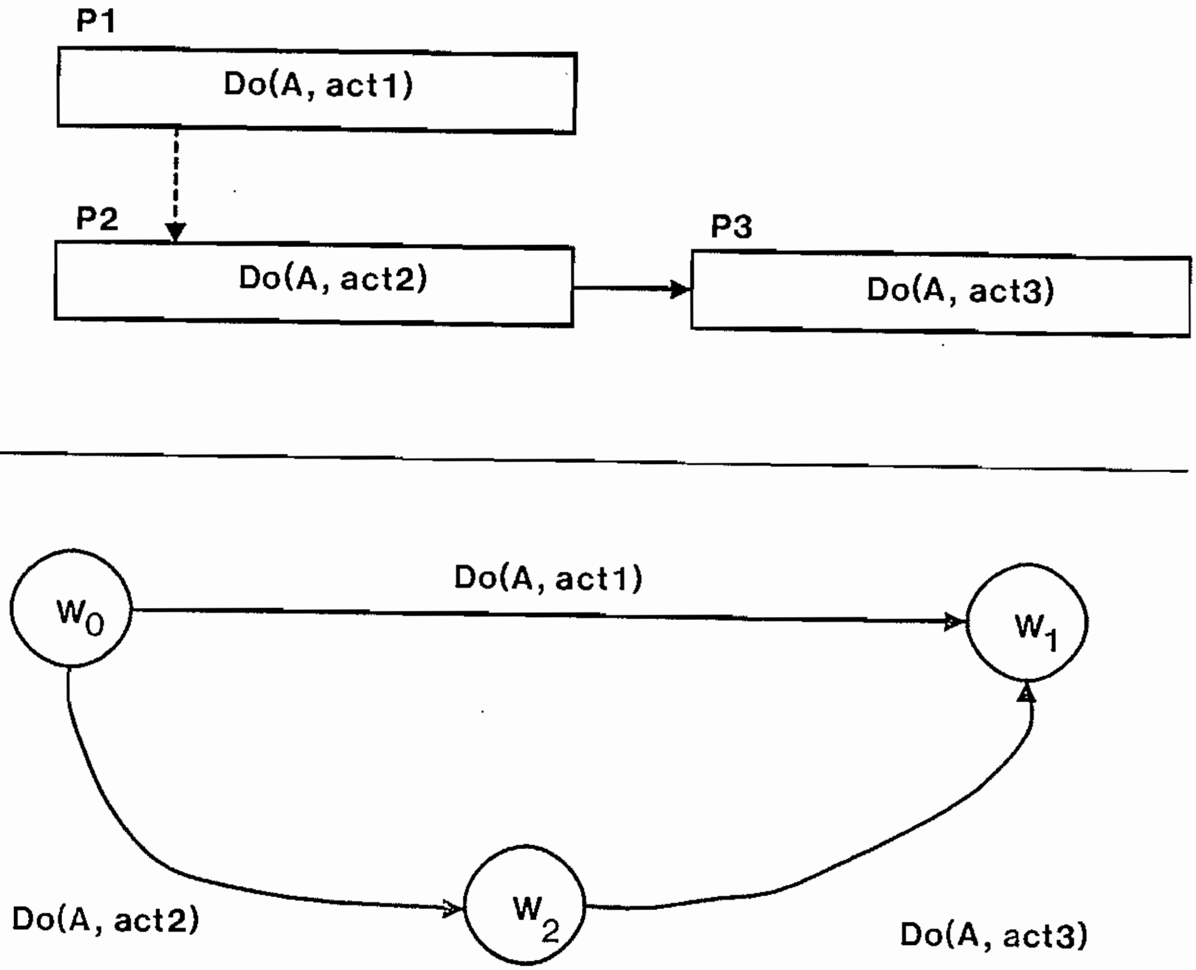

Figure 4.3

Associating Worlds with the Expansion of an Action

action into low-level actions. The world resulting from the execution of the low-level actions is precisely the same world resulting from the execution of the high-level action. If the frame axioms for the high- and low-level actions are carefully designed, it gives one the ability to specify incrementally what aspects of the world stay the same at each level of abstraction.

For example, consider a robot engaged in a block stacking task involving several blocks on a table. Suppose a high-level action of building a tower is proposed. It is conceivable that the block stacking and unstacking operations required to expand tower building to an executable description may make a number of changes in the state of the blocks on the table that cannot be predicted at the time the tower-building action is proposed. However, it is reasonable to assume that no matter what actions are planned as part of that expansion, the position of the furniture in the room would not change. All that can possibly change is the position of blocks on the table top. It is possible to capture this fact in the statement of a frame axiom for the tower-building action.

Using this formalism, the planner can propose a high-level plan and might be able to work on later parts of it without having to expand the initial parts to complete low-level detail. If a situation arises where information is required that depends on the expansion of an earlier part of the plan, the planner can return to the other part of the plan and expand it further before continuing. The ability to state frame axioms for actions at different levels of abstraction is another advantage of KAMP over previous hierarchical planning systems.

5. How KAMP Forms a Plan

KAMP is a multiple agent planning system that forms plans involving cooperative actions among several agents. KAMP's data base contains assertions about what each of the agents know, and what they each know that the other agents know. KAMP is a "third person" planner because it is not actually one of the agents doing the planning, but rather can simulate how the agents would plan, given certain information about them. When KAMP plans, it "identifies" with one of the agents and makes plans from the perspective of the agent it is identifying with. This perspective makes an important difference when the planner considers the wants of other agents. Assuming that an agent A_1 doing the planning has a particular goal to achieve, it is possible for the planner to assume that A_1 will want to do any action that A_1 knows will contribute to achieving the goal. However, if it is necessary to incorporate the actions of another agent, A_2, into the plan, A_1 must be able to show that A_2 will actually do the actions required of him. This amounts to showing that A_2 wants to do the action. Guaranteeing that this condition holds can lead to the planning of requests and commands. Once it is established that A_2 wants to do a high-level action, then the planner assumes that A_2 will want to do any action that he knows will contribute toward the realization of the high-level action. A_2 may not have the knowledge necessary to carry out the action, but it can be assumed that A_2 will execute a plan that he can figure out.

When the planner is given an initial goal, it first creates a procedural network consisting of a single plan step containing the goal. Then the following process is executed repeatedly until either the planner concludes that the goal is unachievable, or some sequence of executable, (i.e., low-level) actions is found that achieves the goal: First, possible worlds are assigned to each of the nodes in the procedural net reflecting the actual state of the world at that time (i.e., at the time *before* the action or goal named in the node is performed or achieved). The initial node is assigned W_0, the initial actual world. Then iteratively, when the planner proposes that a subsequent action is performed in a world to reach a new world, a name is generated for the new world, and an R relation between the original world, the new world, and the action is asserted in the planner's data base. Then all goal nodes that have worlds assigned are evaluated, i.e., the planner calls on the deduction system to attempt to prove that the goal is true using the world assigned to that node as the current state of the actual world. Any goal for which the proof succeeds is marked as a phantom goal.

Next, all the unexpanded nodes in the network that have been assigned worlds, and which are not phantoms, are examined. Some of them may be high-level actions for which a procedure exists to determine the appropriate expansion. These procedures are invoked if they exist, otherwise the node is an unsatisfied goal node, and the action generator is invoked that uses the action summaries to propose a set of actions that might be performed to achieve the goal. If an action is found, it is inserted into the procedural network along with its preconditions, both the universal ones and those specific to the particular action.

Like Sacerdoti's system, KAMP uses procedures called *critics* to examine the plan globally and determine interactions between proposed actions. A critic is a modular procedure that examines a portion of a plan for specific kinds of interactions between actions in the plan. If the interactions occur, the critic reorganizes the structure of the plan in some way.

There is an important distinction between the modifications to the plan made by critics and the modifications made during the process of expanding an action to a lower level of abstraction. The process of expansion is local to an action and concerned with determining what actions can be used to achieve a given goal. It considers only the state of the world as it is assumed to be at the time of performing an action and what actions are available. Critics examine interactions between actions in the plan but do not acutally put actions together to achieve goals. An example is presented in the next section.

The result of separating expansion and criticism is an overall simplification of the planning process. The process of expanding actions is simpler because the many possible interactions do not have to be considered at the time of expansion. Obtaining a rough plan and refining it reduces the amount of blind search the planner has to do. The process of discovering interactions is also simpler because it does not have to be concerned with what actions to perform, only with the interactions between actions that have already been selected.

After the cycle of criticism is completed, the planner checks to see if any goals or high-level actions have been completely expanded to the next level. If the expansion is complete, the planner invokes the deduction system to prove that the proposed sequence of actions actually achieves the goal. If the proof is successful, the process of world assignment is carried out again, and the entire procedure is repeated.

If the proof fails, the planner removes the current choice from the plan and checks to see if other choices can be expanded. The failure of the proof may be due to the inadequacy of the action summaries, and in this case, the planner does not have much better to do than a brute-force search of the search space.

When all the actions at the lowest level of the plan have been expanded as far as possible, the planner moves down to the next lower level of expansion and begins expanding them. If the planner is already at the lowest level and all critics have been applied to the resulting plan, then it has found a complete, executable plan and it returns successfully.

6. An Example of Planning to Affect Knowledge

KAMP and the knowledge representation on which it is based can perhaps best be understood by means of a simple example. Consider the following problem: A robot named Rob and a man named John are in a room that is adjacent to a hallway containing a calendar. Both Rob and John are capable of moving, reading calendars, and talking to each other, and they each know that everyone is capable of performing these actions. They both know they are in the room, and they both know where the hallway is. Neither Rob nor John knows what date it is. Suppose further that John wants to know what day it is, and Rob knows he does. Furthermore, Rob is helpful and wants to do what he can to ensure that John achieves his goal. We would like to see KAMP devise a plan, perhaps involving actions by both Rob and John, that will result in John knowing what day it is.

We would like to see Rob devise a plan that consists of a choice between two alternatives. First, if John could find out where the calendar is, he could go to the calendar and read it, and in the resulting state would know the date. So, Rob might tell John where the calendar is, reasoning that this information is sufficient for John to form and execute a plan that would achieve his goal. The second alternative is for Rob to move into the hall and read the calendar himself, move back into the room, and tell John the date.*

I will not attempt here to make a detailed effort to axiomatize time. Currently, KAMP's temporal reasoning is based on action sequences, and it has no sense of the passing of time other than the occurance of actions. In particular, we will assume that the date does not change during the formulation and execution of the plan to read the calendar.

First we need some basic axioms to describe the state of the world and the possible actions. The date is considered to the the denotation of the term *Date*. Knowing the date is equivalent to knowing the denotation of *Date*. It is universally known that the calendar Cal1 tells the date, so we have the axiom

$$\mathbf{Necessary}(\mathrm{Date} = \mathrm{Info}(\mathrm{Cal1})). \qquad (A1)$$

---

* There are other plans that might conceivably work, like Rob requesting John to come into the hall and then telling him the date, instead of returning to the room. However, to keep things simple, we'll consider only the two alternatives.

where Info(x) is taken to denote whatever information is written on x that can be read by some agent. We need some simple axioms stating the basic facts of the problem:

$$\mathbf{True}(\mathbf{Know}(\text{John}, \text{Location}(\text{Rob}) = \text{Loc1})). \tag{A2}$$

$$\mathbf{True}(\mathbf{Know}(\text{Rob}, \text{Location}(\text{John}) = \text{Loc1})). \tag{A3}$$

$$\mathbf{True}(\mathbf{Know}(\text{Rob}, \text{Location}(\text{Call}) = \text{Loc2}))). \tag{A4}$$

$$\mathbf{True}(\sim \mathbf{KnowsWhatIs}(\text{Rob}, \text{Date})). \tag{A5}$$

$$\mathbf{True}(\mathbf{Know}(\text{Rob}, \sim \mathbf{KnowsWhatIs}(\text{John}, \text{Date}))). \tag{A6}$$

$$\mathbf{True}(\mathbf{Know}(\text{Rob}, \sim \mathbf{KnowsWhatIs}(\text{John}, \text{Location}(\text{Call})))). \tag{A7}$$

$$\forall A\, \mathbf{Necessary}(\mathbf{KnowsWhatIs}(A, \text{Location}(A))). \tag{A8}$$

Three actions can be performed by agents in this domain: moving, informing, and reading. The axiomatization of informing is given in Chapter V. Reading is a type of knowledge producing action that does not involve a speech act, which is axiomatized as follows:

$$\begin{aligned}&\forall A, x, w_1, w_2\, R(\text{:Do}(A, \text{:Read}(x)), w_1, w_2) \supset \\ &\qquad V(w_1, \text{:Location}(A)) = V(w_1, \text{:Location}(x)).\end{aligned} \tag{R1}$$

$$\begin{aligned}&\forall A, x, w_1, w_2\, R(\text{:Do}(A, \text{:Read}(x)), w_1, w_2) \supset \\ &\qquad \forall z V(w_2, z) = V(w_1, z) \wedge \forall P H(w_2, P) \equiv H(w_1, P).\end{aligned} \tag{R2}$$

$$\begin{aligned}&\forall A, x, w_1, w_2\, R(\text{:Do}(A, \text{:Read}(x)), w_1, w_2) \supset \forall w_3 \big[K(A, w_2, w_3) \supset \\ &\qquad \exists w_4\, K(A, w_1, w_4) \wedge R(\text{:Do}(A, \text{:Read}(x)), w_4, w_3) \wedge \\ &\qquad \forall z \big([z = \text{:Info}(x) \supset V(w_3, z) = V(w_1, z)] \wedge \\ &\qquad\qquad [z \neq \text{:Info}(x) \supset V(w_3, z) = V(w_4, z)]\big) \wedge \\ &\qquad \forall P\, H(w_3, P) \equiv H(w_4, P)\big].\end{aligned} \tag{R3}$$

Axioms ($R1$), ($R2$), and ($R3$) look complicated, but it is not difficult to see what they say if one bears in mind the following facts: ($R1$) is a precondition axiom that says that if an agent reads something, he must be in the same place as the object he is reading. ($R2$) describes the physical effects of reading, which is really nothing at all. The axiom says that after an agent reads something, anything true of the world before is also true afterwords, and the values of all functions and constants are unchanged. ($R3$) describes the really important effect of reading, namely that after reading something, an agent knows the value of the expression written on the object.[*]

Moving can be thought of as a strictly physical action whose only knowledge effect is that the agent knows he has just moved. The axiomatization of the action Do(A, Remove(x, y)) is reasonably straightforward and won't be described in detail here. All predicates stay the same, and all terms except the one describing the location of the agent retain the same value. The only precondition is that the agent's starting location is in the initial location x.

For each action, it is necessary to define an action summary. The following are action summaries for the actions used in this problem:

Action:	Do(?A, Inform(?B, ?P))
Preconditions:	**True**(Location(?A) = Location(?B)) **True**(**Know**(?A, ?P))
K-Effects:	**True**(**Know**(?B, ?P))
P-Effects:	None

Action:	Do(?A, Read(?X))

---

[*] Here we are dealing only with reading terms. One could also read object-language predicates, and the treatment of the effects of such an action would be similar to the treatment of object-language predicates in the informing action.

Preconditions: **True**(Location(?A) = Location(?X))
K-Effects: **True(KnowsWhatIs**(?A, Info(?X)))
P-Effects: None

Action: Do(?A, Move(?X, ?Y))
Preconditions: **True**(Location(?A) = ?X)
K-Effects: None
P-Effects: **True**(Location(?A) = ?Y)

For the sake of simplicity, we will assume that "John", "Rob", "Call", "Loc1" and "Loc2" are rigid designators.

KAMP is given the goal **KnowsWhatIs**(John, Date) and is instructed to plan from the perspective of the individual :Rob using world W_0 as the initial state of affairs. The planner creates a single-node procedural network consisting of the given goal.

KAMP first attempts to show that the agent doing the planning (in this case Rob) knows whether the goal is satisfied. If he does not know, then he has to make some sort of plan to find out. To simplify the problem, we assume that Rob already knows that John does not know what time it is (perhaps John just asked Rob for the time) so KAMP does not need to work on this "meta-goal."

KAMP then searches the plan for any high-level actions that need to be expanded and for any unexpanded goal nodes. The current goal node is found, and the action summary list is consulted for actions that have a knowledge-state effect matching

KnowsWhatIs(Rob, Date).

The planner may have to perform a few syntactic manipulations on the goal statement to guarantee the translation of the goal into the meta-language so it will match the effects of the actions stated in the action summaries. In this case it has to know that

$$\mathbf{KnowsWhatIs}(A, P) \equiv \mathbf{Know}(A, P = D(W_0, P)).$$

After performing this transformation, the goal statement matches the knowledge effects of two actions: Inform and Read. The planner knows that John will know what date it is if somebody informs him, or if he finds something that he can read that will tell him the date. Since in our simple axiomatization knowing the date is equivalent to knowing what Cal1 says and since Rob is the only other agent in our environment that can do informing, the plan becomes a choice between either Rob telling John the date, or John reading Cal1.

KAMP creates a choice node to represent the disjunction of these two alternatives and adds the specification that each precondition of the action (including the universal preconditions) be achieved, resulting in the procedural network of Figure 4.4.

KAMP works on expanding each branch of the choice in turn. The first branch is that Rob tells John what time it is. The preconditions for this informing action are that Rob is in the same place as John and that Rob knows what it is that he's informing, i.e., he has to know himself what time it is.* In a manner similar to the previous step, KAMP attempts to show first that

$$\mathbf{Know}(\text{Rob}, \text{Location}(\text{Rob}) = \text{Location}(\text{John}))$$

which follows from the axiom ($A2$) that Rob is in the room (Loc1), axiom ($A3$) that

---

* There are other universal preconditions that could be added, for example, that Rob knows who John is. It is unnecessary to add these preconditions because Rob, John, etc. are rigid designators, and it is assumed that everybody knows who they are. KAMP takes advantage of this fact and only adds explicit universal preconditions for nonrigid terms.

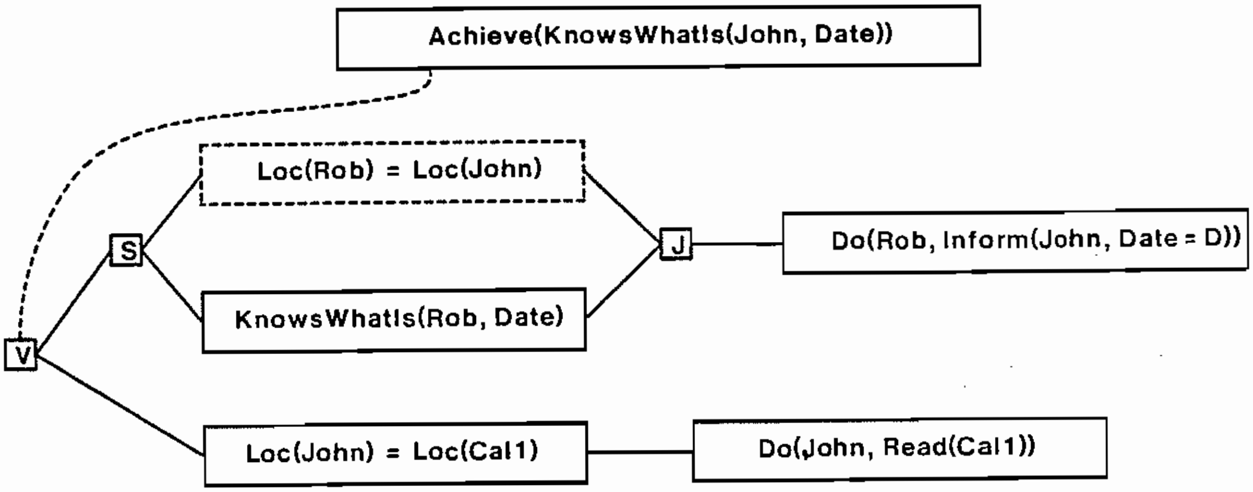

Figure 4.4

Rob Tells John the Time, or John Reads the Calendar

Rob knows John is in the room, and axiom (*A*8) that says in general that everyone always knows where they are. KAMP cannot show that Rob knows what date it is, because it is stated explicitly that he does not, and so therefore a new subgoal is created to achieve that Rob knows what date it is.

Expanding the goal **KnowsWhatIs**(Rob, Date) is done by a process similar to the expansion of **KnowsWhatIs**(John, Date). The action summaries are consulted, and KAMP discovers that Rob will know the date if either somebody tells him the date or he reads the calendar. Since there is only one other agent in our environment, if anyone tells Rob what day it is, it would have to be John. However, this leads to the precondition **KnowsWhatIs**(John, Date) which is already part of the plan we are trying to achieve. KAMP recognizes this circularity and will not

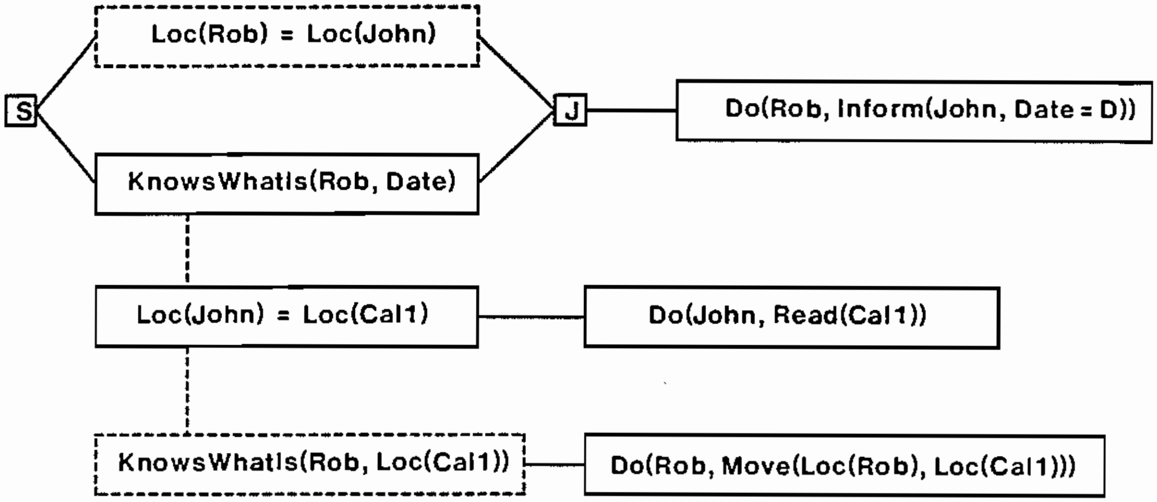

Figure 4.5

Rob Must Be at the Calendar, and Must Read the Calendar

propose that John inform Rob of the date.

Thus, the only action resulting from the expansion of **KnowsWhatIs**(Rob, Date) is that Rob reads Cal1. To do this, Rob must be in the same location as Cal1, and all the universal preconditions must be satisfied. In this case, all that means is that Rob must know what Cal1 is, and that is satisfied because Cal1 is a rigid designator. The expansion results in the procedural net shown in Figure 4.5.

The next cycle of expansion finds the goal that Rob is at the calendar, and this goal is unsatisfied in the current state of the world because Rob is in the room with John. The action summaries give moving as the action to perform to get Rob to a different location, so KAMP plans for Rob to move from Loc1 to Loc2. In the action summary, Loc2 is described intensionally as the location of the thing being read, or in this case Location(Call). This means that the universal precondition is that Rob knows the denotation of the term Location(Call), which is satisfied by axiom ($A4$).

At this point, there are no more goal nodes generated, so in a sense we have a complete plan — it has been expanded down to the lowest level of detail. The plan, however, is incomplete, and if one were to attempt to prove it correct, one would fail. The problem is that once Rob moves out into the hall to read the calendar, he can no longer inform John of the date, because John is back in the room where Rob left him.

Not much has been said about plan criticism up to this point, because until this point in the plan, no critics were applicable. After each cycle of expansion is completed, the critic procedures are invoked. Each critic looks at a very specific condition in the plan, and if the condition obtains, it makes some modifications in the plan that it is hoped will result in some sort of improvement, either in correctness or efficiency.

In this case, there is a critic procedure called *ResolveConflicts* that looks for split nodes in the plan for which all the goal nodes have been expanded on at least one branch of the split. *ResolveConflicts* looks at all the other goal nodes on other branches of the split to see if they are still satisfied after the expanded branch has been executed. If not, an ordering is imposed on the split so that the goal is achieved *after* the expanded branch is executed. KAMP assumes that some such ordering will eventually work. The situation called a "double-cross" where each action undoes the effect or invalidates a precondition of the other (register swapping is a good example of this) is not handled by KAMPand in general presents a difficult problem for hierarchical planners (see Sacerdoti [86]).

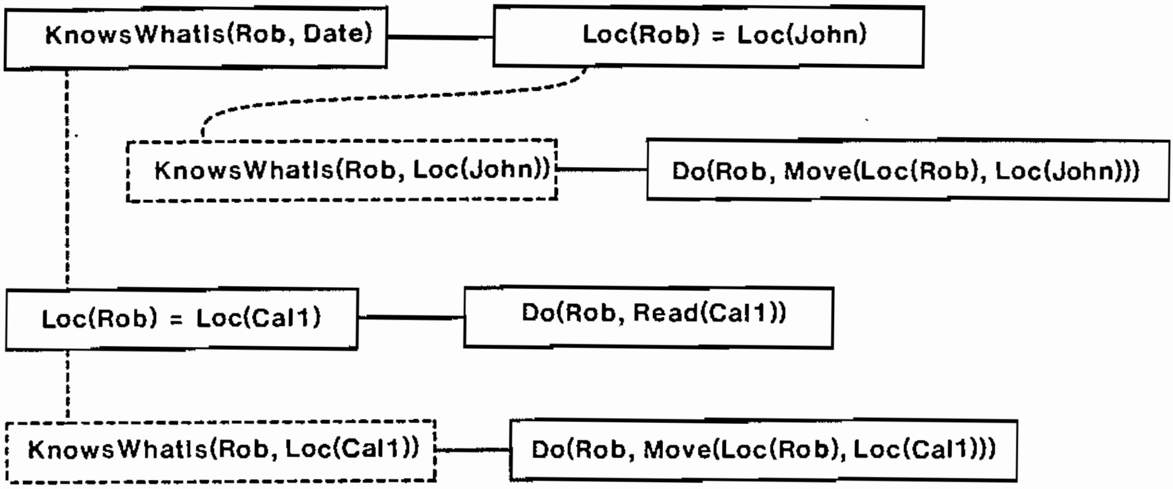

Figure 4.6

After Criticism by ResolveConflicts Critic

In this case, KAMP removes the phantom designation from the goal labeled $G2$ in Figure 4.4, and places the goal after the sequence of actions it has just worked out (see Figure 4.6).

Achieving the goal Location(Rob) = Location(John) is the same as achieving the other location goal, and a move action is planned to get Rob back into the room with John before he performs the informing action.

At this point, the plan has been completely expanded, and no more critics apply, so KAMP tries to verify that it is correct. In this case, the plan can be verified, so no further work is needed.

The expansion of the other alternative to the top-level choice is similar, so it will not be described in detail. KAMP plans for John to move to the location of the calendar, and read it, and in the resulting state John will know the date. Using the universal preconditions, KAMP reasons that for John to move to the calendar, he must know where it is. The only way he can find out where the calendar is is for Rob to tell him, so KAMP incorporates an informing action into the plan to achieve this subgoal.

7. Conclusion

This chapter has discussed several problems in planning to affect the mental state of agents. Chapter III discussed the problems of representing and reasoning about what agents want and believe. It would be desirable for a planning system to make use of the possible worlds formalism for reasoning about how to influence an agent's knowledge. Because planning to affect an agent's knowledge requires reasoning about what he can deduce when some new information is added to his knowledge, in general it is difficult to determine in advance exactly how a given action will affect what he knows. To reduce the amount of search that needs to be done to find a correct plan, action summaries are used to describe common, stereotypical effects of actions on knowledge and the physical world. The planner can use these general heuristics to find a plan that can then be verified to work in the actual situation.

This chapter has introduced the subject of planning to affect some other agent's knowledge. Chapter V considers the planning of illocutionary acts in greater detail, and Chapter VI deals with the problem of producing utterances and how this proccess interacts with the high-level planning processes described here.

V

FORMALIZING AND PLANNING ILLOCUTIONARY ACTS

0. Introduction

This chapter is concerned with the planning of illocutionary acts. It begins with a review of speech-act theory and proposes a set of axioms for the illocutionary acts of informing and requesting that can be used by KAMP in language planning. A basic understanding of how possible-worlds semantics is used to represent a theory of knowledge and action (discussed in Chapter III) is assumed, and the reader is also assumed to be familiar with the general organization of KAMP (described in Chapter IV).

1. What Is a Speech Act?

Speech-act theory has its roots in the work of Wittgenstein, who in *Philosophical Investigations* proposed an analogy between using language and playing games. His basic point was that language is a form of rule-governed behavior, much the same as game-playing, making use of rules and conventions that are mutually known to all the participants.

The field of speech-act theory is usually considered to have been founded by Austin [5] who analyzed certain utterances called performatives. He observed that some utterances do more than say something that is true about the world. In uttering a sentence like, *"I promise to take out the garbage,"* the speaker is not saying anything about the world, but is rather undertaking an obligation. An utterance like, *"I now pronounce you man and wife,"* not only does not say anything that is true about the world, but when uttered in an appropriate context by an appropriate speaker, actually changes the state of the world. Austin argued that the existence of performative utterances required an extension to traditional truth-value semantics.

The most significant contribution to speech-act theory has been made by philosopher John Searle [90][91][92], who developed the first full formulation of the theory of speech acts. The theory can be summarized as follows: Utterances are actions called *illocutionary acts.* These acts fall into several general categories, for example, *directives,* (requests, commands, etc.), *representatives,* (inform, lie, etc.), *commissives,* (promise, threaten, etc.) *expressives,* (apologize, thank, etc.) and *declarations* (utterances that change the state of the world). There are other levels of abstraction* at which an utterance can be viewed, for example, as a series of *utterance acts,* i.e., producing a series of phonemes, or *propositional acts,* which include actions such as referring. Searle analyzed these different categories of speech acts and proposed semiformal sets of conditions under which they may be successfully performed. For example, for each illocutionary act there would be physical enabling conditions and conditions on the beliefs and wants of the speaker that must be satisfied for the action to be performed sincerely and effectively.

---

* KAMP also can view an utterance as a *surface speech act,* which treates the utterance as a linguistic entitiy without regard to deep underlying intentions of the speaker. This is a level of abstraction between that of illocutionary acts and utterance acts, and is described fully in Chapter VI.

Viewed on the intentional level, utterances have two primary components: *illocutionary force* and *propositional content*.* Sentences typically have some means of indicating what speech act the speaker is performing (called an *illocutionary force indicator*) as well as expressing a propositional content. For example, performative utterances have explicit illocutionary force indicators, as in the sentence, *"I hereby order you to take out the garbage."* However, it is much more common to rely upon the syntactic form of the utterance to give a clue as to its illocutionary force, for example, imperative utterances are frequently used to give commands *("Take out the garbage!")*. Finally, there are *indirect speech acts* in which the syntactic form of the utterance does not directly indicate the speaker's intentions. An example is, *"Do you think you could take out the garbage?"* where the speaker intends his question to be understood as a request to take out the garbage.

The effect of successfully performing an illocutionary act is that the hearer acquires some knowledge about the speaker's intentions. For example, if a speaker S informs a hearer H that P by producing an utterance U, then the effect of performing this action is that H knows that S intended to inform H that P, and furthermore intended that this recognition is achieved by means of H's knowledge of the meaning of U. Of course, a speaker may have intentions that go beyond the immediate illocutionary effect, for example, he may intend that H actually believe P, or perhaps intend to make H angry. These effects are sometimes referred to as *perlocutionary effects* and are the major reasons for which speech acts are planned. However, perlocutionary effects are not direct consequences of the speech act, since they depend on the hearer's beliefs and the context in which the act is performed.

---

* Searle points out [91] that not *all* illocutionary acts have propositional content. For example *"Hurrah!"* is an example of an illocutionary act with no propositional content.

Whether or not a particular perlocutionary effect will result from an illocutionary act is something that the planner must reason about during the utterance planning process.

The term "speech act" is often imprecise because it is not clear what level of abstraction is being addressed. In some sense, all utterances are "speech acts." Throughout this thesis, "illocutionary act" will be used to refer to speech acts at their highest level of abstraction. Illocutionary acts are actions such as informing, requesting, and promising. Illocutionary acts are realized by virtue of performing utterance acts. If the utterance acts are chosen with proper consideration of the conventions of the language and the hearer's knowledge, then the illocutionary act will be successfully realized.

Stating the effects of illocutionary acts in terms of the hearer's recognition of the speaker's intentions is important, because the process of understanding an utterance frequently requires interpreting the speaker's intentions behind the action. Allen [1], [2] designed a language-understanding system (or perhaps more appropriately, an illocutionary-act interpreter, since it did not actually interpret surface sentences) that would interpret illocutionary acts in the light of what it knew about the speaker's intentions. For example, if a speaker asks the attendant at an information booth, *"Where is the train to Montreal?"* the system would infer that the speaker probably wanted to meet the train when it came in, so it would respond by furnishing information about both the time and place of its arrival, since that would maximally facilitate what it believed to be the hearer's plan. Allen claims that understanding underlying intentions is the key to interpreting indirect speech acts such as, *"Do you know what time it is?"*

From a theoretical standpoint, it is also important that the hearer believe that the speaker wants to convey his intentions through the hearer's understanding of the meaning of the utterance. This condition may seem obvious, but ignoring it can lead to problems. For example, consider a situation in which I want to impress someone by making them believe I am a fluent speaker of French. I could just inform him by saying, *"I speak French,"* but another way to bring about this belief would be to say some utterance that the hearer believes to be in French, although he may not understand its literal meaning. Therefore, I could cause the hearer to believe that I speak French by uttering some nonsense like, *"La plume de ma tante est sur la table."* It is odd to classify this utterance as a normal illocutionary act because its intended effect has no relation to the meaning of the utterance. Since any French utterance would be adequate for the purpose, I could have just as well have said, *"Je parle français,"* which literally means, "I speak French." In this case, I have caused the hearer to believe that I speak French by producing an utterance that literally means "I speak French," but this case is really no different from the case where I uttered nonsense. One must conclude that to successfully perform an illocutionary act, the hearer must recognize the intention of the speaker by means of understanding the meaning of the utterance.

2. The Relationship between Illocutionary Acts and Utterances

At first glance, it may seem that there is a direct correspondence between illocutionary acts and utterances. A speaker will plan an illocutionary act such as INFORM(H, P), and then to realize the INFORM, he utters a declarative sentence with propositional content P. Unfortunately the situation is not quite so simple, because the speaker has many options for realizing the INFORM, only some of which involve the utterance of a sentence with propositional content P. For instance, instead of realizing the inform directly with a declarative sentence, the speaker may elect to realize it indirectly by way of a question. It may also be possible for the speaker to realize the informing action by modifying another utterance already planned for another purpose, without any sentence planned explicitly to realize the INFORM.

Because of the need for some intermediate level of abstraction between the level of illocutionary acts and the utterance of a series of words, *surface speech acts* are defined. Cohen and Levesque [17] defined similar actions to provide a formal means of terminating the intention recognition process, but did not apply it to multiple-effect utterances. Surface speech acts are abstractions for the actions of producing particular kinds of sentences. The kinds of sentences under consideration for English would be declarative, interrogative, and imperative sentences — the primary mood choices. The surface speech acts corresponding to these choices are called respectively DECLARE, ASK, and COMMAND.

Surface speech acts also provide a convenient level of abstraction for describing the effects of an utterance on the discourse focus and are discussed in greater detail in Section 4. It is important to remember that illocutionary acts are abstract communicative acts and there is *no* simple one-to-one correspondence between illocutionary acts and utterances.

3. Formalizing Illocutionary Acts

One of the central problems of language planning is devising a formalism for illocutionary acts that both captures the essence of what it means to perform an illocutionary act and that also is sufficiently straightforward so that a planner can reason with it efficiently.

The first attempt at such a formalization was made by Cohen [15]. Cohen's formalization of illocutionary acts is a reasonably straightforward rendition in logic of Searle's conditions for the successful performance of various illocutionary acts [90]. Cohen divided his preconditions into two groups: *want preconditions* involving conditions on the speaker's wants, and *can do* preconditions, which covered all other prerequisites. The effects of illocutionary acts were formalized as the hearer knowing that the speaker wants the hearer to believe something or do something. To bridge the gap between the illocutionary effect and the intended perlocutionary effect, Cohen proposed formal "actions" that would accomplish that purpose. For example, if the goal was $\mathbf{Believe}(H, P)$, Cohen's planner would plan an illocutionary act $\mathrm{Do}(S, \mathrm{Inform}(H, P))$ that would produce as an effect

$$\mathbf{Believe}(H, \mathbf{Want}(S, \mathbf{Believe}(H, P))). \tag{$E1$}$$

Since it is impossible for a speaker to directly influence a hearer's beliefs, Cohen proposed a formal action called CONVINCE that represented the process of the hearer accepting the proposition of the speaker's utterance as true. CONVINCE has $(E1)$ as a precondition and produces the desired hearer belief as the effect. The CONVINCE action is somewhat ad-hoc because there is no identifiable action that the speaker performs that realizes it. Such an "action" is not necessary in a system that is based on a sufficiently powerful formalism to draw conclusions about when an agent will believe something given that he knows that some other agent believes it.

In later work, Cohen and Levesque [17] place the burden of intention recognition on *surface speech acts* by proposing a planning formalism with operators like S-INFORM and S-REQUEST, which are surface realizations of INFORM and REQUEST. The surface speech acts are intended to correspond to utterances with a given mood, for example, S-INFORMs are declarative sentences. The effect of an S-INFORM is formalized as the speaker and hearer mutually believe that the speaker wants the hearer to know that he believes some proposition, i.e., that

$$\mathbf{MutuallyBelieve}(S, H, \mathbf{Want}(S, \mathbf{Believe}(H, \mathbf{Believe}(S, P)))).$$

The basic idea of this approach is incorporated into KAMP because it is necessary to cut off the intention-recognition process by formalizing actions as directly producing recognition of intention. The level of the surface sentence is an ideal point to make this cut-off for two reasons. First, speakers of the same language will mutually know a large variety of conventions about their language, and they know that as long as they use the conventions of the language appropriately, it will be guaranteed that their intentions will be interpreted correctly by others. In general it is impossible for a speaker to say P and intend $\sim P$, except in cases of irony or sarcasm, but even in those cases the speaker usually provides intonation and other clues to clearly signal his intentions. Second, and perhaps more important, it is difficult to describe the effects of lower level linguistic actions such as the utterance of a word so their effects are independent of the context in which the actions are performed. Describing recognition of intention at this level would be difficult and, in my opinion, would probably not lead to an elegant or even satisfactory theory.

The problem that the hearer is faced with upon hearing an utterance is to decide what illocutionary act is being performed by the speaker. If the speaker is behaving according to the conventions of the language, this process will be relatively straightforward. We will exclude from consideration here cases in which a speaker performs one illocutionary act and intends the hearer to recognize another, such as performing a lie and intending the hearer to recognize it as an INFORM. Indirect speech acts are covered by this analysis, although they can sometimes be viewed as an instance of the speaker performing one illocutionary act and intending the recognition of another. Indirect speech acts are discussed in greater detail in the next section.

In axiomatizing illocutionary acts, as with axiomatizing any facts about the world, it is necessary to choose some level of detail of description that both captures the essential properties of the concepts that one wishes to reason about while avoiding detail that will unnecessarily complicate reasoning in the limited set of cases that are expected to arise. With illocutionary acts, this decision amounts to assigning the role of recognition of intention in the speech-act understanding process. Entirely eliminating recognition of intention simplifies the planning process, but limits the system's flexibility to deal with certain kinds of situations such as indirect speech acts. On the other hand, reliance on recognition of intention gives the system much flexibility and more closely models the performance of humans, but greatly complicates the reasoning processes.

The first and most obvious path to follow is to simply declare that the result of an informing action such as Do(S, Inform(H, P)) is simply **Believe**(H, P). This axiomatization involves no recognition of intention. In spite of its simplicity and obvious shortcomings, such a simple description of illocutionary acts can be adequate in a surprisingly large number of situations. For example, in task-oriented dialogues in which an expert with much domain knowledge is assisting an apprentice with relatively little knowledge, the apprentice usually believes what the expert says, since he has no reason to believe he is being misled. Similarly, the expert always believes the apprentice is making sincere requests. This simple analysis breaks down when one wants to model a situation in which the hearer does not necessarily believe anything the speaker says. An example of a simple situation in which this applies is if one wants to state a rule such as, "A judge will believe a witness' testimony if he knows that the witness was at the scene of the crime."

A modification to the simple proposal that results in the ability to reason about whether or not an assertion will be believed by the hearer is to define the effects of the informing act $\mathrm{Do}(S, \mathrm{Inform}(H, P))$ as

$$\mathbf{Believe}(H, \mathbf{Believe}(S, P)),$$

i.e., the hearer knows that the speaker believes P. This allows one to state axioms about when one agent believes something that he knows another agent believes.

A further refinement is to include the recognition of intention in the definition of the illocutionary act. The effect of $\mathrm{Do}(S, \mathrm{Inform}(H, P))$ is

$$\mathbf{Believe}(H, \mathbf{Want}(S, \mathbf{Believe}(H, P))).$$

This definition facilitates plan recognition, since the hearer, after knowing that the speaker wants him to believe P, is led naturally to the question of how the hearer's belief that P facilitates the speaker's plan.

One of the desirable features of the KAMP system is that these different levels of axiomatization of illocutionary acts can be combined to the overall advantage of the system. Action summaries are based on the simpler effects, and the more complex effects involving intention recognition are described by the axioms used by the deduction system to reason about how the world has changed after an action has been performed. Since a large number of common cases will be covered by the basic actions encoded in the action summaries, the process of verification will often succeed with no problems. When it does not, clues are provided by the failed proof tree about what went wrong and how to correct the defficiency.

In the illocutionary act formalism proposed here, it is assumed that the speaker and hearer mutually know what illocutionary act has been performed after the speaker performs some surface speech act that conforms with the conventions of the language. As is the case with the other actions described in Chapter III, the preconditions and effects of illocutionary acts are assumed to be universal knowledge. The formalization of INFORM is similar in form to that proposed for physical actions in Chapter III. Several axioms are needed: one to state the preconditions of informing, one to state the physical effects of the action, one to state the effects of the action on the speaker and hearer's mutual knowledge, and a "knowledge state frame axiom" to describe the effect on the knowledge of other agents that may be unaware that the action has taken place. As is the case when describing the knowledge effects of physical actions, the knowledge effects of illocutionary acts can be deduced from general world knowledge and the implicitly represented fact that all agents know what it means to do informing.

In the following axioms, A and B are the speaker and hearer, respectively, w_1 is the world in which the action is performed, w_2 is the world resulting from the performance of the action, and P is a variable ranging over object-language terms. Axiom ($I1$) describes the preconditions of informing:

$$\begin{aligned}\forall A, B, P, w_1, w_2\, R(&\text{:Do}(A, \text{:Inform}(B, P)), w_1, w_2) \supset \\ &V(w_1, \text{:Location}(A)) = V(w_1, \text{:Location}(B)) \wedge \\ &T(w_1, \mathbf{Want}(@(A), \mathbf{Know}(@(B), @(P)))) \wedge \\ &T(w_1, \mathbf{Know}(A, P)).\end{aligned} \tag{$I1$}$$

Axiom ($I1$) says that if A informs B that P, then A and B must be at the same location (a physical enabling condition), A must want B to know P, and A must know himself that P is true (sincerity condition).

It is assumed here that informing (and the performance of illocutionary acts in general) does not alter the physical state of the world. Therefore, informing has no physical effects, and frame axioms state that everything that is true before the action will also be true after the action, and that the values of all terms remain the same. This is captured by axiom ($I2$):

$$\begin{aligned}&\forall A, B, P, w_1, w_2, R(\text{:Do}(A, \text{:Inform}(B, P)), w_1, w_2) \supset \\ &\qquad \forall Q\, H(w_1, Q) \equiv H(w_2, Q) \wedge \forall x, V(w_1, x) = V(w_2, x).\end{aligned} \tag{$I2$}$$

Surprisingly, the axiom that describes the knowledge effects of INFORM is very simple, since all it needs to state is that the speaker and hearer mutually know the action has taken place. Axiom ($I3$) is essentially the same as the axioms of the knowledge effects of actions such as reading and moving, described in Chapter IV.

$$\begin{aligned}&\forall A, B, P, w_1, w_2\, R(\text{:Do}(A, \text{:Inform}(B, P)), w_1, w_2) \supset \\ &\qquad \forall w_3\, K(\text{Kernel}(A, B), w_2, w_3) \supset \exists w_4\, K(\text{Kernel}(A, B), w_1, w_4) \wedge \\ &\qquad R(\text{:Do}(A, \text{:Inform}(B, P)), w_4, w_3).\end{aligned} \tag{$I3$}$$

Given the precondition axiom ($I1$), it is possible to deduce that after A has performed the informing action, B knows that A wants him to know P and that A himself knows that P.

In addition to the above axioms ($I1$), ($I2$), and ($I3$), an action summary for KAMP must be written. The action summary will reflect the physical preconditions of the action, the basic knowledge state preconditions, and will state as the effect of the action that B knows P.

The axiomatization of REQUEST is quite similar to the axiomatization of INFORM, and as was the case with INFORM, there are several levels of detail of intention recognition that one could choose. It will be assumed that a REQUEST always involves some future action of the hearer. Therefore, the arguments to REQUEST are the intended hearer and an intensional description of the action. The simplest description of REQUEST states that the effect of A_1 requesting A_2 to do P is that A_2 wants to do P. This suffers from the same problem that the oversimplified definition of INFORM did, namely that it allows no possiblity for A_2 to refuse the request. A more realistic axiomatization would have as its effect $\mathbf{Know}(A_2, \mathbf{WantsToDo}(A_1,\mathrm{P}))$. In this case one also needs some sort of "helpfulness axiom" that will allow one to conclude $\mathbf{WantsToDo}(A_2, P)$ from $\mathbf{Know}(A_2, \mathbf{WantsToDo}(A_1, P))$. Of course, it is possible, and occasionally desirable, to carry the intention-recognition process one step further and describe the effect of requesting as

$$\mathbf{Know}(A_2, \mathbf{WantsToDo}(A_1, \mathbf{Know}(A_2, \mathbf{WantsToDo}(A_1, P)))),$$

but this will not be required for any of the examples described here.

The assertion $\mathbf{Want}(A_1, P)$ is a reasonable sincerity condition for A_1 to request P of some other agent. Since this is universally known, the knowledge-state effects of REQUEST are described similarly to that of other actions in the possible-worlds formalism. The set of axioms required for REQUEST are as follows:

Preconditions:

$$\begin{aligned}&\forall A, B, P, w_1, w_2\ R(\text{:Do}(A, \text{:Request}(B, P)), w_1, w_2) \supset\\ &\qquad V(w_1, \text{:Location}(A)) = V(w_1, \text{:Location}(B)) \land \qquad (R1)\\ &\qquad T(w_1, \mathbf{WantsToDo}(@(A), @(P))).\end{aligned}$$

Physical effects:

$$\begin{aligned}&\forall A, B, P, w_1, w_2\, R(\text{:Do}(A, \text{:Request}(B,P)), w_1, w_2) \supset \\ &\qquad \forall z\, V(w_2, z) = V(w_1, z) \wedge \\ &\qquad \forall Q\, H(w_2, Q) \equiv H(w_1, Q).\end{aligned} \tag{R2}$$

Knowledge-state effects:

$$\begin{aligned}&\forall A, B, P, w_1, w_2\, R(\text{:Do}(A, \text{:Request}(B,P)), w_1, w_2) \supset \\ &\qquad \big[\forall w_3\, K(\text{Kernel}(A,B), w_2, w_3) \supset \\ &\qquad\quad \exists w_4\, K(\text{Kernel}(A,B), w_1, w_4) \wedge \\ &\qquad\qquad R(\text{:Do}(A, \text{:Request}(B,P)), w_4, w_3)\big].\end{aligned} \tag{R3}$$

Helpfulness axiom:

$$\begin{aligned}&\forall A, B, P, w\, T(w, \text{Helpfully-Disposed}(A,B)) \wedge \\ &\qquad \big[\mathbf{Know}(A, \mathbf{WantsToDo}(B,P)) \supset T(w, \mathbf{WantsToDo}(A,P))\big].\end{aligned} \tag{R4}$$

The axioms $(R1)$ through $(R4)$ provide the knowledge needed to draw conclusions about agent B's wants after A performs a request.* The alert reader may notice that axiom $(R4)$ will cause some difficulty for most deduction systems. If $\mathbf{Want}(A,P)$ is a goal and $(R4)$ is used in a backward direction, the resulting subgoal will be $\exists x\, \mathbf{Know}(A, \mathbf{Want}(@(x), P))$, and since x can be bound to B, attempting to prove $\mathbf{Know}(A, \mathbf{Want}(B,P))$ will eventually lead to the subgoal $\exists x\, \mathbf{Know}(A, \mathbf{Know}(B, \mathbf{Want}(@(x), P)))$. This recursive subgoal will keep turning up over and over again, each time embedded in one more level of A and B's knowledge. This recursion can be detected and broken by syntactic restrictions on the application of the rule.

---

* Some details about additional axioms covering mutual knowledge, A's knowledge and wants, and knowledge and wants of agents other than A and B have been surpressed, since they add complexity to the example without providing much enlightenment.

4. Conclusion

This chapter has shown how illocutionary acts can be axiomatized within Moore's possible-worlds-semantics formalism for reasoning about knowledge and action, and that the resulting axiomatization can be used efficiently by KAMP to generate plans. The key idea was to axiomatize illocutionary acts as actions that produce the knowledge that they have been performed. This, together with conditions on the speaker's knowledge and intentions, also expressed by the axioms as preconditions enable the hearer to reason about what the speaker wants and knows.

Action summaries provide a simpler level of description of the same action that heuristically facilitates the generation of plans involving illocutionary acts. The next chapter on planning of surface linguistic actions describes how the illocutionary acts described in this chapter can be realized as actual utterances.

VI

PLANNING SURFACE LINGUISTIC ACTS

0. Introduction

This chapter discusses the problems of planning surface linguistic actions, including *surface speech acts, concept activation,* and *focusing.* Since it is possible to describe a linguistic action on the illocutionary level without commiting oneself to any particular strategy for its realization, these linguistic actions are at a lower level of abstraction in the action hierarchy than the illocutionary acts discussed in Chapter V.

The planning process that produces surface linguistic acts is different from that producing the more abstract actions, because at this level grammatical constraints enter into the planning process. Many grammatical constraints, when viewed from a planning perspective, are completely arbitrary. For example, as far as a planner is concerned, there is no obvious reason for the syntactic requirement of English that adjectives preceed nouns in a noun phrase. Any attempt to force such a constraint to depend on the speaker's goals (excluding, of course, the goal of producing coherent English) is bound to fail. Planning at the level of surface linguistic acts consists of the combination and expansion of illocutionary acts according to the rules of the grammar of the language. When a modification is to be made to the plan, the planner must check that the modification will be allowed by the constraints imposed by the language.

1. The Role of Grammatical Knowledge

The grammar employed by KAMP is not the traditional grammar consisting of a set of rules describing all and only the legal syntactic structures of the language. With KAMP, grammatical decisions must be made by a variety of procedures with a narrow jurisdiction, such as the expansion procedures for illocutionary acts, or the critics that test for a particular kind of global interaction. Therefore, instead of being localized in one set of rules, the grammatical knowledge is spread throughout the system in the expansion procedures and critics of the planner. When one of the planning procedures desires to make a modification to the plan, it has enough grammatical knowledge to decide whether the proposed modification is acceptable or not. For example, the procedure that expands the surface speech act for declarative sentences has some grammatical knowledge that describes the syntactic structure of English declaratives, including passives and datives. A critic that may later propose adding another case argument to a sentence has grammatical knowledge concerning when such an addition is possible, depending on the choice of verb and the set of syntactic structures it can accomodate. The expansion process that plans noun phrases has procedurally encoded grammatical knowledge describing the structure of English noun phrases.

The utterance syntax tree is associated with the surface speech act node in the plan and is used as a working data structure by the planner, since the structural relationships between constituents in a sentence are better represented by a tree than by the sequencing relationships most naturally represented in a procedural net. Whenever a surface speech-act node is added to the plan, a syntax tree is created that reflects the basic syntactic features of the sentence. This tree grows and evolves as the plan develops, and linguistic actions are expanded to greater detail. The tree is annotated to show the relationship between portions of the tree and parts of the procedural net because modifications to the plan require modifications to the syntactic structure of the sentence, and vice versa.

The grammatical knowledge is represented as conditions and actions within the planning modules that have responsiblilty for making particular grammatical decisions. This is not a particularly perspicuous way to represent a grammar, and it is a weakness of KAMP that it does not have access to an independent grammar whose linguistic merit can be judged independently of the performance of the program. An independent grammar would be useful for the following reasons: (1) the linguistic competence of the system could be characterized apart from running the program and seeing what it does, (2) the grammar would be better organized, enabling the author of the grammar to more easily modify the system and predict the effects and interactions resulting from the changes. Neither of these desirable features bears directly on the primary motivation for KAMP, which is to describe how illocutionary acts are realized as utterances and to account for how speakers achieve multiple goals in a single utterance. Therefore, the representation of the grammar has been assigned secondary importance in this research.

2. Surface Speech Acts

Surface speech acts were introduced in Chapter V to serve as an abstract representation of an utterance. There is a one-to-one correspondence between surface speech acts and utterances since the former are merely abstract representations of the latter. No such correspondence holds between illocutionary acts and utterances.

A surface speech act is only *one* possible strategy for the expansion of an illocutionary act to the next lower level of abstraction. However, it is the most important one because it is impossible to realize an illocutionary act without either designing a surface speech act to realize it or incorporating the action into some surface speech act that is being planned to realize another illocutionary act. Therefore, any plan that involves the planning of some illocutionary acts must necessarily involve the planning of at least one surface speech act.

Corresponding to the three basic syntactic mood choices in English, there are three types of surface speech acts. The surface speech act COMMAND is realized by imperative sentences, ASK by interrogative sentences, and DECLARE by declarative sentences.

The effect of a surface speech act is that the speaker and hearer mutually believe that the illocutionary act realized by the surface speech act has been performed. For example, if a speaker realizes an INFORM by planning a DECLARE of some proposition, the effect of the DECLARE is that the speaker and hearer mutually believe that an INFORM-that-the-proposition-is-true has taken place.

It is impossible to state simple axioms describing the effects of surface speech acts in the same manner as has been done for illocutionary acts for two reasons: (1) the same surface speech act can realize different illocutionary acts depending on the context, and (2) it is possible for a surface speech act to realize several illocutionary acts. A surface speech act could realize one action in one context and several actions in another, given a different set of speaker and hearer beliefs.

Some standard indirect requests are best described as a choice of the surface speech act to realize a request. For example an ASK action can be planned to realize a REQUEST to perform a salt-passing action in the sentence *"Can you pass the salt?"* although it can be regarded as a REQUEST to INFORM the hearer whether he has the ability to perform a salt-passing action. (See Section 6 of this chapter for a more thorough discussion of indirect speech acts and implicatures.) Any axiomatization of ASK would have to account for the difference in effect of the utterance in different contexts. It is axiomatizable in principle, but one seems to have little to gain from such an effort.

Section 5 on action subsumption describes how an utterance like, *"Tighten the screw with the long Philips screwdriver."* can realize several illocutonary acts, like a REQUEST to tighten the screw and an INFORM that the tool for tightening the screw is the long Philips screwdriver. Given that the speaker knows that the hearer doesn't know that a particular screwdriver is a Philips screwdriver, the utterance could in that case also serve to inform the hearer that the long screwdriver is a Philips screwdriver. This is contrasted with the case where "long" is used to distinguish long versus short. So, not only is it the same surface speech act can realize different types of illocutionary acts in different contexts, but it can realize a different number of illocutionary acts in different situations.

Since the effects of a surface speech act are not stated explicitly in a context independent manner, KAMP assumes that the effects of a surface speech act are a conjunction of the effects of the illocutionary acts the surface speech act has been planned to realize. Formally, the planner treats the surface speech act as a single low-level action that "expands" a number of higher level actions. This is different from the usual situation in hierarchical planning in which several low-level actions are usually required to expand a high-level action to the next level of abstraction. The world resulting from the performance of the surface speech act is treated as being identical to the world resulting from the performance of each of the illocutionary acts in an arbitrarily chosen sequence, as illustrated in Figure 6.1.

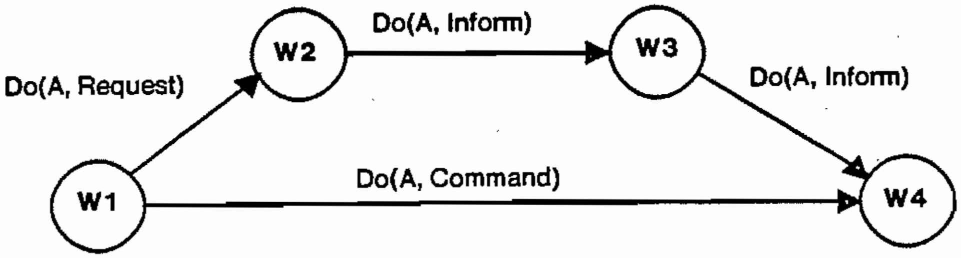

Figure 6.1

Worlds Related by a Surface Speech Act Realizing Multiple Illocutionary Acts

This can be compared with the usual case illustrated in Figure IV.3.

Representing the relationship between illocutionary acts and surface speech acts in procedural networks also presents some minor difficulties. Problems arise in situations in which one low-level action serves as the expansion of several high-level actions. This is an instance of true parallelism, and it is reasonable to think of the performance of a surface speech act as executing several illocutionary acts in parallel. However, the KAMP formalism is not adapted to describing parallel actions. KAMP treats the surface speech act as the expansion of one of the illocutionary acts and marks the other actions as being *subsumed* by the surface speech act. The subsumed actions have a pointer to the surface speech act that subsumes them, as illustrated in the procedural net in Figure 6.2.

Chapter VII describes a detailed example of the planning of an utterance and describes in detail how KAMP treats the interaction between illocutionary acts and surface speech acts.

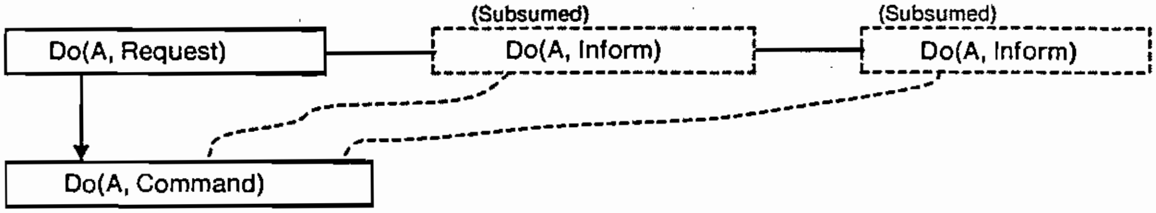

Figure 6.2

A Surface Speech Act in a Procedural Net

3. Planning Concept-Activation Actions

The next lower level of abstraction below surface-speech acts is that of *concept activation actions*, which are generalized referring actions. The term "concept" means some object language term that denotes an individual in the real world.

Traditionally, reference is a semantic concept. Terms in some language, be it natural or formal, refer to objects in the world. There is a great deal of philosophical literature on reference and denotation. Early theories, such as Russell's required that for an expression in natural language to refer (definitely) to some object A, the expression must embody some predicate $\mathcal{P}$.[*] According to this analysis, the definite noun phrase, "the red book", refers to a particular book, $B1$, if and only if

$$\forall x \, \mathrm{Book}(x) \wedge \mathrm{Red}(x) \supset x = B1,$$

---

[*] Throughout this thesis, script letters are used as a schema to represent a formula that may consist of several terms and involve other variables.

ignoring information about context, or focus.**

The problem with attempting to define reference precisely for natural languages is that the relationship described in ($R1$) frequently does not hold. This has led speech act theorists like Strawson [97] and Searle [91] to distinguish between speaker reference (i.e., what the speaker intends to communicate) and semantic reference (i.e., what the utterance refers to objectively, without regard to speaker intentions.) Ignoring such obvious deficiencies as ($R1$) does not take any discourse or pragmatic knowledge into account, Kripke [53] gives examples in which speakers often plan referring expressions that succeed as far as the hearer is concerned, but do not satisfy ($R1$) because the description is not true of the objective world. The classic example is the case in which two speakers are talking about another man at a party, and one says to the other, "The man holding the martini ..." and is understood perfectly well, even though the man he intended to refer to was in reality holding a glass of water. It is possible to follow this principle in constructing arbitrarily complicated examples as in [79].

To circumvent problems of this sort, we will not talk about natural-language expressions as referring to anything. Natural-language expressions can be mapped into an intensional logical form, and one can then talk about the denotation of terms in the logic. We have adopted an intensional object language (described in Chapter III) that is ideal for this purpose. A sentence like, "The man holding the martini is a spy," can be represented as

$$\iota x{:}(\mathrm{Man}(x) \land \text{Holding-Martini}(x))\,\mathrm{Spy}(x).$$

** KAMP is capable of using information about context and focus in planning referring expressions, but a discussion of these problems is deferred until the next subsection.

where the notation $\iota x{:}\mathcal{P}(x)\mathcal{Q}(x)$ is intended to mean the formal equivalent to the statement, "The x such that $\mathcal{P}(x)$ has property $\mathcal{Q}$." The expressions $\mathcal{P}$ and $\mathcal{Q}$ can contain modal operators, and since the ι operator is a special type of existential quantifier, it is possible to have

$$\mathbf{Know}(A, \iota x{:}P(x)Q(x))$$

as well as

$$\iota x{:}P(x)\mathbf{Know}(A, Q(x)).$$

This turns out to be a very convenient notation for describing the logical form of sentences. It is always true that

$$\iota x{:}\mathcal{P}(x)\mathcal{Q}(x) \equiv \exists x\, \mathcal{P}(x) \wedge \mathcal{Q}(x) \wedge \left[\forall y\, \mathcal{P}(y) \supset x = y.\right]$$

It is possible to axiomatize the "man and martini" example using the possible worlds formalism we have adopted using ($A1$) and ($A2$) as follows:

$$\exists x\, H(W_0, \text{:Man}(x)) \wedge H(W_0, \text{:Holding-Martini}(x)) \supset V(W_0, x) = \text{:Man1} \qquad (A1)$$

and

$$\exists x\, H(W_0, \text{:Man}(x)) \wedge H(W_0, \text{:Holding-Water}(x)) \supset V(W_0, x) = \text{:Man2}, \qquad (A2)$$

i.e., in the real world, Man1 is the man holding the martini and Man2 is holding a glass of water, and that

$$\begin{aligned}&\exists x\, \forall w\, B(\text{Kernel}(S, H), W_0, w) \supset \\ &\qquad \left[H(W_0, \text{:Man}(x)) \wedge H(W_0, \text{:Holding-Martini}(x)) \supset V(w, x) = \text{:Man2}\right]\end{aligned}$$

or, that the speaker and the hearer mutually believe that there is a man holding a martini, and he is Man2. This axiomatizes the critical part of the example.

The verb "to refer" is used differently by different people. Some philosophers and logicians speak of terms referring to objects in the world. Reference in this sense is strictly a semantic concept and has nothing to do with actions performed by speakers. In this sense, reference is the same as denotation. When speech act theorists talk about a speaker referring to an object A, they often mean that a speaker performs an action that can be construed as the utterance of a term that denotes A. Cohen and Perrault carry this concept further saying that a speaker refers by uttering a term that the hearer interprets as an attempt by the speaker to get the hearer to realize that the speaker wants to refer to A, where the actual denotation of the term uttered is somewhat problematical (see [79]). This is the sense in which the word "refer" is intended in this thesis.

The action called *concept activation* captures this notion of referring at a sufficiently high level of abstraction so that it is not constrained to be a purely linguistic action. When a concept-activation action is expanded to a lower level of abstraction, it can result in the planning of a noun phrase within the surface speech act of which the concept activation is a part, *and* also physical actions such as pointing that also communicate the speaker's intention to refer. It is this potential nonlinguistic component that distinguishes concept activation from referring, which is a purely linguistic action.

Concept-activation actions introduce new intensional concepts. For example if a speaker activates the concept $\iota x{:}P(x)$, then a new individual is introduced, say

$G0015$,[*] and $P(G0015)$ is asserted. It is possible that according to the hearer's knowledge, $G0015$ can be shown to be equal to some individual who is already known to exist. On the other hand, it may not be possible, and from this point on in the dialogue, the speaker can continue activating the concept $G0015$ without knowing what individual it refers to.

Speakers can signal through their choice of utterances whether the speaker is believed to know the referent of the term introduced by a concept activation. One common method is by the use of definite or indefinite determiners. An indefinite determiner means that the speaker does not intend that the hearer find a referent. A definite determiner may or may not signal such an intention, depending on how the speaker is using the description.

Concept Activation and Planning Descriptions

Concept-activation actions usually lead to the planning of some description. A *description* $\mathcal{D}$ of an individual is a conjunction of predicates, each of which is true of the individual:

$$\forall x\, \mathcal{D}(x) \equiv \mathcal{P}_1(x) \wedge \ldots \wedge \mathcal{P}_n(x),$$

where predicates in a script font (such as $\mathcal{D}$ and $\mathcal{P}$) are object language predicates that apply to their argument and perhaps other functions and free variables as well. Each of the $\mathcal{P}_i$ is a *descriptor.* A description $\mathcal{D}$ is adequate for a speaker S to activate concept C for hearer H if it is true that (ignoring focusing for the time being)

$$\textbf{MutuallyBelieve}(S, H, \forall x\, \mathcal{D}(x) \supset x = C) \qquad (C1)$$

---

[*] To suggest its similarity to a GENSYM atom. One could think of *G0015* as an "object language skolem constant."

In the case of indefinite and attributive reference, the description is already specified as part of the concept to be activated. For example, activating the concept $\iota x{:}P(x)$ constrains the speaker to use the description P. If a speaker has a goal that the hearer hold a belief about a particular individual, then the representation of that goal must necessarily involve the use of a rigid designator for that object. Rigid designators are part of the logic used to describe utterances, but there is no such thing as a rigid designator in natural language. Therefore, when a speaker plans an utterance in which he wants to refer to some particular individual A, he constructs a description of A that he believes the hearer can identify as intended by the speaker to correspond to A.

At first it may seem that condition ($C1$) is somewhat strict, since the condition involves mutual belief instead of just requiring that the speaker and hearer believe the description holds. Clark and Marshall [12] point out how it is possible to construct examples for which the speaker's concept activation fails if the mutual belief condition is not met. Cohen and Perrault [79] show that the mutual belief condition is too strong, but for a different reason. They construct examples based on the "man and martini" example cited earlier, in which the speaker and hearer believe that the description is *false*, but the speaker and hearer succeed at referring successfully as long as at some level the description is mutually believed, in other words, that the speaker believes that the speaker and hearer mutually believe the description, or the speaker believes that the hearer believes it is mutually believed, etc.

Mutual belief and mutual knowledge present problems for deduction. It is difficult to prove that two agents mutually know something unless it is already known that they do because the definition of mutual knowledge in terms of conditions on knowledge about knowledge requires the verification of an infinite number of conditions. Clark and Marshall's solution is that speakers use *copresence heuristics* to draw conclusions about what is mutually believed. It is assumed that all speakers that have enough in common to communicate at all share a great deal of knowledge from their cultural background, their current physical situation, and the history of the dialogue they have engaged in. For example, if two agents are looking at a table with some blocks on it and they mutually know they can see, then they can conclude that they both mutually know the color, size, and location of all the blocks on the table. The example presented in Chapter VII gives more detailed information about how KAMP uses mutual knowledge in planning descriptions.

The Expansion of Concept-Activation Actions

The planning of a concept activation is similar to the planning of an illocutionary act in that the speaker is trying to get the hearer to recognize his intention to perform the act. This means that all that is necessary from a high-level planning point of view is that the speaker perform some action that signals to the hearer that the speaker wants to call the hearer's attention to some object. This is commonly done by incorporating a description of the object into the utterance, but there is no real requirement that this attention-getting action be a linguistic one. Any action that is interpreted by the hearer as the speaker's attempt to call his attention to something would suffice. For example, the speaker could point at an object (clearly a communicative act), or perhaps throw it at the hearer (not so clearly communicative, but attention-getting).

The problem is that concept-activation actions are planned during the course of the expansion of surface speech acts. This means that actions that occur in the expansion have to be linguistic acts, or at least in most cases have a linguistic component. A speaker cannot point at a rock and say, "— *is my pet rock.*" The speaker is forced to perform some sort of linguistic action regardless of the means chosen to communicate his intention. Therefore, all concept activations are planned with two components, an intention-communication component and a surface-linguistic component. The intention-communication component consists of the action or strategy chosen by the speaker to communicate his intention to refer. The surface-linguistic component consists of the realization of the intentional component, taking into consideration the grammar. The speaker can activate a concept by planning a set of mutually believed descriptors that uniquely describe the object, as described previously. The surface-linguistic component for this choice consists of examining the predicates chosen for the description and the grammatical options for realizing them and attempting to find an expression (usually a noun phrase) that incorporates all the chosen predicates. Instead of planning a description, the speaker can choose to perform some physical action (like pointing) that will communicate his intentions to the hearer. The surface-linguistic component specifies what linguistic actions are to be coordinated with the speaker's physical ones, for example the use of deictic determiners like "this" and "that" while pointing.

Formalizing Concept Activation

Concept-activation actions are formalized in a manner similar to illocutionary acts. They are formalized as having a direct effect on the speaker's and hearer's mutual knowledge of what the current active concept is. It is assumed that describing and pointing are low-level communicative actions that do not require an explicit account of the hearer's recognition of the speaker's intention to activate a concept.

The fact that pointing doesn't have any effect on the object pointed to makes it easier to analyze as a nonlinguistic communicative act that establishes the speaker's intention to activate a concept. Of course, agents can perform other physical actions that could be interpreted as emobdying a speaker's intention to refer, as well as satisfying other goals, for example, throwing a ball and saying, "Catch this," which simultaneously satisfies the speaker's goal of moving the ball and activating a concept. However, such complexities are beyond the scope of the examples under consideration here.

The axiomatization of concept activation is described by axioms ($A1$), ($A2$), and ($A3$). Axiom ($A1$) describes the simple precondition that the speaker and hearer have to be at the same location; ($A2$) describes the effect. (Note that it does not state what does *not* change because that information is discovered during the expansion of the concept activation into low-level actions of describing or pointing.) The concept :Active(A, B, C) holds in a world when the concept C is active with respect to speaker and hearer A and B. Axiom ($A3$) is the standard effect that the hearer knows the action has been performed.* The function describing the action is :Cact(H, C) where H is the hearer, and C is the concept to be activated.

$$\begin{aligned}&\forall A, B, C, w_1, w_2\, R(\text{:Do}(A, \text{:Cact}(B, C)), w_1, w_2) \supset \\ &\qquad\qquad V(w_1, \text{:Location}(A)) = V(w_1, \text{:Location}(B)).\end{aligned} \tag{A1}$$

$$\forall A, B, C, w_1, w_2\, R(\text{:Do}(A, \text{:Cact}(B, C)), w_1, w_2) \supset H(w_2, \text{:Active}(A, B, C)). \tag{A2}$$

---

* Three more axioms are needed, almost identical to ($A3$) to describe the effect of the action on the speaker's knowledge, on the speaker's and hearer's mutual knowledge, and on the knowledge of agents other than the speaker and hearer, but this has been omitted here because it is not necessary for a conceptual understanding of the situation.

$$
\begin{aligned}
&\forall A, B, C, w_1, w_2\, R(\text{:Do}(A, \text{:Cact}(B, C)), w_1, w_2) \supset \\
&\qquad \forall w_3, K(B, w_2, w_3) \supset \\
&\qquad \exists w_4\, K(B, w_1, w_4) \wedge R(\text{:Do}(A, \text{:Cact}(B, C)), w_4, w_3).
\end{aligned}
\tag{A3}
$$

Axioms ($D1$), ($D2$), and ($D3$) are a formal axiomatization of the describe action. The function :Describe($B, C, \mathcal{D}$) is intended to mean the action of describing the concept C to hearer B using description $\mathcal{D}$. The description $\mathcal{D}$ is assumed to be a conjunction of object language predicates that are applied to C. Since the axioms as stated are not in first-order logic, they are not used by KAMP's deduction system exactly as stated. However, the equivalent knowledge is used by the procedure that expands concept-activation actions, as described in Chapter VII during the discussion of the example. Axiom ($D1$) gives the precondition that the description is known to be true of its referent by the speaker and that the speaker and hearer mutually believe that the description picks out the referent. Axiom ($D2$) says that the only thing that changes after uttering a description is what is active, and ($D3$) states that the speaker knows in the resulting situation that the describe has been performed.

$$
\begin{aligned}
&\forall A, B, C, \mathcal{D}, w_1, w_2\, R(\text{:Do}(A, \text{:Describe}(B, C, \mathcal{D})), w_1, w_2) \supset \\
&\qquad T(w_1, \mathbf{Know}(A, \mathcal{D}(C))) \wedge \\
&\qquad T(w_1, \mathbf{MutuallyKnow}(A, B, \forall x\, \mathcal{D}(x) \supset x = C)).
\end{aligned}
\tag{D1}
$$

$$
\begin{aligned}
&\forall A, B, C, \mathcal{D}, w_1, w_2\, R(\text{:Do}(A, \text{:Describe}(B, C, \mathcal{D})), w_1, w_2) \supset \\
&\qquad \forall x\, V(w_1, x) = V(w_2, x) \wedge \\
&\qquad \forall y, z\, [y \neq \text{:Active}(A, B, z) \supset H(w_1, y) \equiv H(w_2, y)].
\end{aligned}
\tag{D2}
$$

$$\begin{aligned}&\forall A, B, C, D, w_1, w_2\, R(\text{:Do}(A, \text{:Describe}(B, C, D)), w_1, w_2) \supset\\ &\qquad \forall w_3\, K(B, w_2, w_3) \supset\\ &\qquad \exists w_4\, K(B, w_1, w_4) \wedge R(\text{:Do}(A, \text{:Describe}(B, C, D)), w_4, w_3).\end{aligned} \tag{D3}$$

The axioms for pointing are similar to those for describing, except for the preconditions. It may seem odd that such different actions are described with the same effects, but all we are really trying to capture are the effects of the action on the hearer's knowledge.

$$\begin{aligned}&\forall A, B, x, w_1, w_2\, R(\text{:Do}(A, \text{:Point}(B, x)), w_1, w_2) \supset\\ &\qquad H(w_1, \text{:HandEmpty}(B)) \wedge\\ &\qquad V(w_1, \text{:Location}(A)) = V(w_1, \text{:Location}(B)) \wedge\\ &\qquad V(w_1, \text{:Location}(A)) = V(w_1, \text{:Location}(x)).\end{aligned} \tag{P1}$$

$$\begin{aligned}&\forall A, B, x, w_1, w_2\, R(\text{:Do}(A, \text{:Point}(B, x)), w_1, w_2) \supset\\ &\qquad \forall x\, V(w_1, x) = V(w_2, x) \wedge\\ &\qquad \forall y, z\, y \neq \text{:Active}(A, B, z) \supset H(w_1, y) \equiv H(w_2, y).\end{aligned} \tag{P2}$$

$$\begin{aligned}&\forall A, B, x, w_1, w_2\, R(\text{:Do}(A, \text{:Point}(B, x)), w_1, w_2) \supset\\ &\qquad \forall w_3\, K(B, w_2, w_3) \supset\\ &\qquad \exists w_4\, K(B, w_1, w_4) \wedge R(\text{:Do}(A, \text{:Point}(B, x)), w_4, w_3).\end{aligned} \tag{P3}$$

4. Axiomatizing the Effect of Utterances on Discourse Focus

Focusing is a natural part of any communication process. When two agents participate in a dialogue, they share some mutual knowledge of what is being discussed. These mutual beliefs can arise from general mutually held knowledge of the topic of the discourse (see Grosz [32]) or from specific linguistic cues that the speaker uses to inform the hearer of what he intends to focus on (see Sidner [93] and Reichman [81]). Such cues can take the form of *clue words* such as "anyway", "by the way," "next," "then," etc. or in the choice of marked syntactic structure, such as cleft, pseudocleft, and topicalized sentences. (See Creider [19] for an explanation of the focusing rules associated with different marked syntactic structures.)

Since one of the intentions that a speaker communicates to a hearer is what he intends to focus on, it is natural that focusing should play an important role in the language-planning process. During the planning of an extended discourse, the speaker will discover situations in which it is important to communicate the intention to shift focus, and he may plan a high-level *focusing action* to satisfy the focusing goal. Although it may be possible to perform global focusing actions with physical actions such as pointing, such actions will frequently have a linguistic component and will be subsumed by surface speech acts. Section 5 describes how this action subsumption process works.

The problem with the axiomatization of Reichman's topic-shifting actions along the same lines as Sidner's focusing rules is that it is difficult to formalize some of the intuitive notions that Reichman deals with, in particular, the general notion of a discourse being "about" something. Reichman partitions dialogues into *context spaces,* but although it is reasonably clear to speakers of the language what a context space is, it is difficult to capture this intuitive notion formally. Grosz's and Sidner's focusing algorithms make use of similar notions that are sufficiently restrictive to be handled formally, but are not sufficiently general to describe what happens to a speaker and hearer's mutual belief when a clue word is uttered.

Because of the inadequacy of the formal tools currently available, the problem of planning intentional focus shifting must be left for future research. However, it is possible to encode Sidner's focusing rules in the formalism that has been chosen for KAMP, and KAMP can use these rules in the generation of definite descriptions and pronominal references.

Sidner devised a set of rules for tracking the movement of focus as a sequence of utterances in a discourse are understood. The rules specify the new focus as a function of the previous foci, the objects of previous noun phrases, the syntactic structure chosen, and consistency with general world knowledge. The algorithm will not be described in detail because it is fully specified in [93]. Sidner's algorithm is designed for an understanding system, but it is reasonably straightforward to adapt it to generation as well.

In addition to the algorithm for tracking the focus, Sidner proposes a number of rules for using the knowledge about discourse focus to interpret anaphora, such as definite noun phrases and pronominal reference. These rules can also be adapted to generation to decide how to refer to something that is already believed to be in focus.

There are three issues to be decided before incorporating focusing into KAMP: (1) the focusing predicates must be defined, (2) it must be decided what actions change the focus, and (3) the focusing information must be used by KAMP to generate referring expressions. Sidner's focusing algorithm is designed for tracking the object in immediate focus, so a predicate called *ImmedateFocus* is used to apply to an *intensional* description of this object. This intensional description is a conjunction of object language predicates that are specified by the concept-activation action to be used in the referring description. Thus, it is not necessary for the participants to know what a description denotes in order for the referent to be in focus, as long as they mutually believe that it denotes the same individual. Since a focusing mechanism requires a stack that can be pushed and popped, the *ImmediateFocus* predicate applies to both the intensional concept and the "stack pointer" of the current focus. It is important in formalizing focus movement to describe which entities are possible next foci for the discourse. The designation of potential focus results from some concept-activation action being performed in a previous sentence. The concept-activation introduces a concept as a new potential focus, and a subsequent concept-activation signals the movement of focus to the new concept. This is how the focus moves in most situations, unless the speaker chooses a marked syntactic structure (e.g., a pseudocleft sentence) specifically for moving the focus.

Since the state of the focus depends on the syntactic structure of the utterance, the most reasonable place to describe the effects of focusing is on the level of surface speech-acts. Once a particular syntactic structure has been chosen, all the information needed to deduce what will happen to the focus has been specified, and the focusing effects are asserted by the surface speech-act expansion process. Finally, the process that generates descriptions for concept-activation uses the latest mutually believed focusing information together with Sidner's rules for pronoun selection to generate a pronominal reference where appropriate.

The Problem of Lexical Choice

The final step in the expansion of a concept activation is the insertion of the actual words into the syntax tree of the associated surface speech-act. This is a complicated problem for which KAMP has only a simple and inadequate solution.

It is clear that speakers can satisfy additional goals through the lexical realization of the descriptions. Often these goals concern difficult-to-formalize concepts of attitude and politeness. For example, the words "film" and "movie" could both be chosen to realize a "motion-picture" predicate, but for many speakers, the former conveys a more culturally refined and dignified attitude.

KAMP assumes that there is a straightforward correspondence between the predicates in its logical representation and words in its lexicon. Often there will be several words that realize a given predicate, but the criteria for choosing between them will involve considerations like those outlined in the "buy versus sell" example cited earlier, rather than attitude and politeness.

5. Subsumption of Linguistic Actions

An action A_1 *subsumes* another action A_2 if A_1 and A_2 are part of the same plan and action A_1, in addition to producing the effects for which it was planned (i.e., the principal effects) also produces the effects for which action A_2 was intended. Therefore, the resulting plan need only include action A_1 to achieve all the goals.

During the course of planning linguistic actions, many options are available to the planner for constructing utterances. Frequently, the planner can detect situations where minor alterations in one of the actions will result in an action that subsumes an action in another part of the plan. The term 'minor alterations' is somewhat vague, but the general idea is clear. When planning surface speech-acts, it means making a change localized to only one of the constituents of the sentence. Changes can be made to a surface speech-act during the course of planning that do not alter the overall structure of the utterance, but are sufficient to subsume other actions in the plan. Examples of such changes are adding a descriptor to the description in a concept activation, adding nonrestrictive relative clauses to noun phrases, and conjunction.

Action subsumption is an excellent example of a global interaction between actions in a plan. It is for the detection and resolution of such interactions that *critics* are introduced into the hierarchical planning process. Chapter IV discussed some of the critics used by KAMP, for example the *ResolveConflicts* critic that detects and resolves destructive interations between the effects of actions in parallel branches of conjunctive splits. The *ActionSubsumption* critic is more complicated than the standard language-independent critics because it has much more information to consider. It first has to detect the possibility of subsumptions, which requires the knowledge of what kinds of relationships must hold between actions before subsumption rules can apply, and then it must know what alterations must be made to the subsuming action to make the subsumption successful.

It is not always possible for one illocutionary act to subsume another just because they both refer to common concepts. For example, Sidner [93] pointed out that in cases in which the concept is already in focus, the normal subsumption strategy does not work. Consider the following examples:

Harold bought a book from the Stanford Bookstore. (*S1*)
? *The green book was autographed by the author.* (*S2a*)
* *The book that was autographed by the author was green.* (*S2b*)
The green tome was required for his physics class. (*S2c*)

Sentence ($S1$) could be followed by ($S2a$), ($S2b$), or ($S2c$). Sentence ($S2a$) sounds a little strange, since when "book" is in immediate focus, the hearer expects the speaker to refer to it with a pronoun. Postnominal modifiers make it even more difficult for a noun phrase to cospecify the focus, so ($S2b$) is found to be unacceptable to most speakers. However, ($S2c$) is acceptable to most speakers, since the speaker uses a different lexical item that entails the same properties as the focus. The *ActionSubsumption* critic must detect the focusing constraint and propose a primary descriptor that will be interpreted correctly by the hearer.

Cohen* has also observed that the modality of the conversation affects the amount of action subsumption that people do when planning utterances. Action subsumption occurs more frequently in dialogues over teletype links than in face to face contact. It is speculated that either the greater difficulty of teletype communication motivates planning more efficient communication, or the increased quantity of time available allows more complex planning processes to take place. KAMP makes no attempt to explain the processing constraints that contribute to human decisions whether to subsume actions, but this is an interesting topic for research.

6. Planning Indirect Speech Acts

Some utterances are intended by the speaker to have an illocutionary force other than their obvious surface meaning. Such utterances are called indirect speech acts, of which sentences ($E1$) and ($E2$) are examples:

Do you want to play some backgammon? (*E1*)

It's two o'clock and I have to work. (*E2*)

Sentence ($E1$) is a question in it's surface form, but the speaker obviously intends the hearer to recognize it as a request to actually play a game of backgammon. ($E2$) is a refusal by the speaker to comply with the hearer's request, but the refusal, rather than a simple "No" is realized by the speaker by informing the hearer that he

---

* Personal communication.

has to work. The hearer is expected to know that having to work precludes playing backgammon and therefore the speaker intends the hearer to recognize a refusal. Searle makes the point [91] that indirect speech acts are intended literally, but the underlying illocutionary act that the speaker intends for the hearer to recognize entails the proposition expressed in the surface speech act. Thus, when the speaker asks, "Could you pass the salt?", it is acceptable for the speaker to answer the question literally (e.g., "Yes, here it is," or "No, I can't reach it.") as long as the intention to make a request is recognized. Clark [13] has performed experiments that seem to indicate that speakers do process and respond to the surface form of indirect requests in addition to recognizing the underlying intentions.

Planning indirect speech-acts is important because they arise frequently in natural discourse, and it is an important mechanism by which speakers achieve multiple goals through utterances. KAMP does not currently plan indirect speech acts, not because it is inherently incapable of doing so, but rather because the types of goals that are generally satisfied through the use of indirect speech acts involve concepts such as politeness that are difficult to formalize. Statements like "leave options" and "don't impose" have to be defined precisely enough to permit some formal treatment. (Lakoff [54] gives examples of the relevant considerations that need to be formalized.)

Searle [91] lists some rules about how speakers can perform indirect commissives. For example one rule is that "[a speaker] can make an indirect commissive by either asking whether or stating that the preparatory condition concerning his ability to do [an action] obtains." Brown [8] has extended these rules to a variety of speech acts, including requesting and informing, and has used these rules as the basis of a system to recognize and interpret indirect speech acts.

KAMP could be extended to plan indirect speech acts by first planning illocutionary acts without considering interactions with other goals as described in Chapter V. The plan would also contain suitably expressed goals of conveying the degree of politeness appropriate to the given situation. During the criticism cycle, a critic would notice the co-occurrence of an illocutionary act such as REQUEST and a politeness goal. The critic would propose satisfaction of the politeness goal by appropriate expansion of the illocutionary act. When the illocutionary act is expanded into a surface speech-act, the expansion procedure would consult its rules about indirect speech-act conventions (such as specified by Brown [8]) and then propose an indirect realization, using the indirect conventions as a heuristic. Then, during the verification cycle, the planner would check to make sure that the speaker knows that his intentions will be correctly recognized by the hearer.

7. Conclusion

This chapter has examined several issues pertaining to the planning of surface linguistic acts. It has always been stressed in this thesis that utterances are multifaceted actions that produce many kinds of effects simultaneously. A single utterance can inform the hearer of several propositions, make a request, change the speaker's and hearer's beliefs about the focus, and inform the hearer about the speaker's social view of the hearer. The language planner's task is to plan actions that satisfy goals along each of these dimensions and then to realize these high-level actions as utterances (and perhaps physical actions as well) in the most efficient manner possible. This chapter has discussed how the KAMP language planning system constructs the surface form of an utterance that satisfies multiple goals.

Surface linguistic acts are near the bottom of the abstraction hierarchy of linguistic actions, just above the production of words. Surface speech-acts are utterances viewed as an abstract, partially specified syntactic structure. Concept activation actions are abstract referring actions that expand into the utterance of a particular description as well as physical actions to signal the speaker's intention to refer.

The planning of efficient actions requires the planner to recognize when *action subsumption* is possible and to take appropriate steps to incorporate multiple high-level actions into a surface speech act. Much of the planner's linguistic knowledge is directed towards knowing when such combinations are possible. The ability to recognize and perform action subsumption is the key to KAMP's ability to produce appropriate utterances to achieve its goals.

VII

AN IMPLEMENTED EXAMPLE OF PLANNING AN UTTERANCE

0. Introduction

This chapter discusses in detail an example that requires KAMP to form a plan involving several physical and illocutionary acts and then to integrate the multiple illocutionary acts into a single utterance. Many details of the planning process, it is hoped, will be made clear that could only be alluded to in Chapter VI. It is important to realize that the implementation of KAMP was done to test the *feasability* of a particular approach to multiple-agent planning and language generation. It is not intended to be a "production" system, and, for this reason, many details of efficiency have been overlooked.

KAMP is based on a first-order logic natural-deduction system* that is similar in many respects to the one proposed by Moore [74]. The current implementation does not take advantage of well-known techniques that are the topics of much recent research in the design of theorem provers, such as structure-sharing or indexing, for example. However, the system is reliable, if not efficient, at making the necessary deductions to solve problems similar to the one described here.

The entire KAMP system is implemented in INTERLISP-D on a Xerox Dorado,**

---

* The credit for the initial implementation of the deduction system belongs to Mabry Tyson.

and the example discussed in this chapter requires about 40 minutes to run to completion. Without apologizing further for the implementation inadequacies, I will admit that the time required to produce a single utterance almost certainly precludes the practical application of this approach in the near future. However, the theoretical ideas are important, since it is apparent from the examination of dialogues between people that reasoning processes similar to those modeled by KAMP must be undertaken by speakers during the production of utterances. It is clear from this research that modeling these reasoning processes requires a great deal of computational power given the deduction system. It remains a topic for future research to determine how the ideas presented in this thesis can be applied at practical costs.

1. The Problem and the Domain

KAMP's initial domain is the information that is required by an expert system that knows about the assembly and repair of a particular piece of equipment, and that knows that the user is a novice seeking assistance. There are two reasons for choosing this particular domain: first, dialogue protocols have been collected (e.g., Deutsch [20]) that provide a body of linguistic data raising interesting issues and examples of phenomena that can be explained by the theory on which KAMP is based, second, the domain provides an ideal situation for multiple-agent planning in which communicative actions arise naturally.

Figure 7.1 illustrates a typical situation in which KAMP operates. This domain

---

** The Xerox Dorado is an experimental single-user computer system designed at the Palo Alto Research Center roughly comparable to a DEC KL-10 in speed. INTERLISP-D is a version of INTERLISP implemented on the Dorado that exploits features of the machine such as a large address space. KAMP requires an address space larger than the 18 bits of a DEC 10 or 20 series machine to run the example described in this chapter.

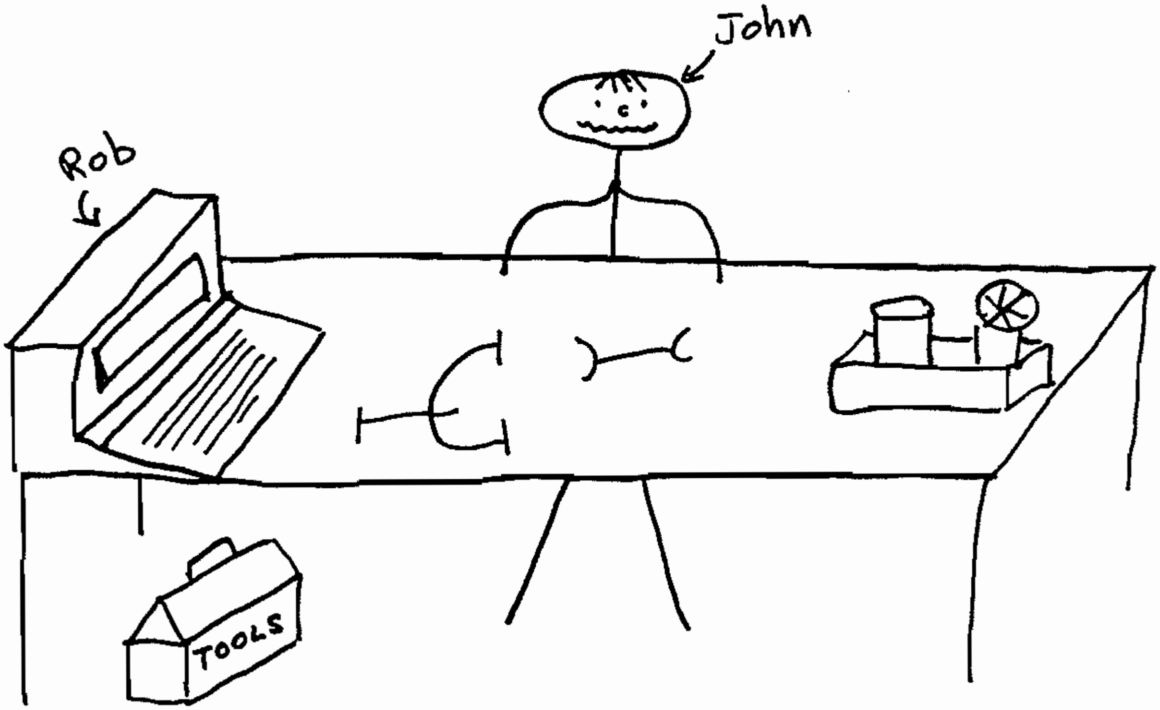

Figure 7.1

KAMP's Domain

has two agents called Rob and John. Rob is a robot that incorporates KAMP for planning and deduction. Rob's only connection with the world is the computer terminal, so he is capable of performing speech acts, but no actions that directly affect the physical state of the world. John is assumed to be a person who is capable of performing both speech acts and physical actions. The particular situation for this example includes a piece of equipment to be repaired (in this case an air-compressor) and some tools that are necessary for the task. The tools can be out in plain view on the table, in which case Rob and John both know their location, or they can be stored away out of sight in the tool box, in which case Rob may know where they are, but not necessarily John. In general, Rob is the expert, and he knows almost everything about the situation. For example, Rob knows how to assemble the compressor because he knows how the parts fit together, he knows what tools to use for the various assembly operations, and he knows where all the tools are located.

This domain provides an ideal setting for studying multiple agent planning as it relates to the production of utterances. Communication arises naturally in this domain because of the difference in knowledge and capabilities of the agents. Since Rob is incapable of performing physical actions, he must make requests of John whenever he wants to change the physical state of the world. Since Rob knows all there is to know about the task and John knows this, John must ask questions to get the information he needs to do a task, and Rob must provide John with the information he knows he needs when he requests John to do something. Therefore, the need for communication arises for either agent to satisfy his goals.

Part of the description of the domain includes an axiomatization of the possible actions that can be performed by the agents and the corresponding KAMP action summaries. The initial state of the physical world must be described, as well as each agent's knowledge about the world.

The following assertions describe the initial state of the world in the example under consideration (the symbols John, Rob, PU, PL, T1, TB1, WR1, B1, LOC1 and LOC2 are all rigid designators):

$$\mathbf{Necessary}(\mathrm{Human}(\mathrm{John})) \tag{A1}$$

$$\mathbf{Necessary}(\mathrm{Robot}(\mathrm{Rob})) \tag{A2}$$

$$\mathbf{Necessary}(\mathrm{Pump}(\mathrm{PU})) \tag{A3}$$

$$\mathbf{Necessary}(\text{Platform}(\text{PL})) \tag{A4}$$

$$\mathbf{Necessary}(\text{Table}(\text{T1})) \tag{A5}$$

$$\mathbf{Necessary}(\text{Tool-box}(\text{TB1})) \tag{A6}$$

$$\mathbf{Necessary}(\text{Wrench}(\text{WR1})) \tag{A7}$$

$$\mathbf{Necessary}(\text{Box-end}(\text{WR1})) \tag{A8}$$

$$\mathbf{Necessary}(\text{Bolt}(\text{B1})) \tag{A9}$$

$$\mathbf{Necessary}(\mathbf{KnowsWhatIs}(\text{Rob}, \text{Location}(\text{John}))) \tag{A10}$$

$$\mathbf{Necessary}(\mathbf{KnowsWhatIs}(\text{John}, \text{Location}(\text{Rob}))) \tag{A11}$$

$$\mathbf{Necessary}(\forall x\, \text{Wrench}(x) \supset \text{Tool}(x)) \tag{A12a}$$

$$\mathbf{Necessary}(\forall x\, \text{Human}(x) \vee \text{Robot}(x) \supset \text{Animate}(x)) \tag{A12b}$$

$$\begin{aligned}\mathbf{Necessary}(\forall x, y, z\, \text{Pump}(x) \wedge \text{Attached}(x, y) \wedge \\ \text{Attached}(x, z) \supset y = z)\end{aligned} \tag{A13}$$

$$\mathbf{Necessary}(\forall x\, \text{Animate}(x) \supset \mathbf{KnowsWhatIs}(x, \text{Location}(x))) \tag{A14}$$

Notice that since axioms ($A1$)-($A14$) are *necessarily* true (i.e., true in all possible worlds), they are universally known. It may seem implausible that the facts expressed by axioms ($A10$) and ($A11$) should be treated as necessary truths, but they will be for the purpose of simplifying the example. We shall assume that Rob and John always know each other's location, regardless of any moving actions that may take place. The following facts are true about the world, but are not necessarily true, since they change over time:

$$\mathbf{True}(\text{Location}(\text{John}) = \text{LOC1}) \tag{A15}$$

$$\mathbf{True}(\text{Location}(\text{Rob}) = \text{LOC1}) \tag{A16}$$

The following assertions describe the mutual knowledge of the agents:

$$\mathbf{True}(\mathbf{MutuallyKnow}(\text{John}, \text{Rob}, \text{Attached}(\text{PU}, \text{PL}))) \tag{A17}$$

$$\mathbf{True}(\mathbf{MutuallyKnow}(\text{John}, \text{Rob}, \text{Fastener}(\text{B1}, \text{PU}, \text{PL}))) \tag{A18}$$

$$\mathbf{True}(\mathbf{MutuallyKnow}(\text{John}, \text{Rob}, \text{Fastened}(\text{B1}, \text{PU}, \text{PL}))) \tag{A19}$$

$$\mathbf{True}(\mathbf{MutuallyKnow}(\text{John}, \text{Rob}, \text{Location}(\text{TB1}) = \text{LOC2})) \tag{A20}$$

$$\textbf{True}(\textbf{MutuallyKnow}(\text{John}, \text{Rob}, \text{Location}(\text{PL}) = \text{LOC1})) \qquad (A21)$$

The example also requires some axioms that describe the instrument relation:

$$\forall x, y, z, i\, \text{Fastener}(z, x) \wedge (\text{Tool}(z) = i) \supset \text{Instrument}(\text{Unfasten}(x, y, z, i)) \qquad (A22)$$

$$\forall x, y, z, i\, \text{Instrument}(\text{Unfasten}(x, y, z, i)) \supset \text{Instrument}(\text{Remove}(x, y, i)) \qquad (A23)$$

Axiom ($A23$) says that the instrument of an unfastening action is also the instrument of a removing action, which is natural, since unfastening is part of the process of removing. Axiom ($A22$) says that if z is some fastener (e.g., a bolt) that attaches x to something, and i is an appropriate tool for manipulating it (e.g., a wrench), then i is an instrument for any action of unfastening x from whatever it is attached to.

The domain-specific axioms are completed by some axioms that describe the knowledge of agents that is *not* shared between them. In this case, we will assume that Rob knows that the tool for removing the bolt is the wrench WR1, and that it is located in the tool-box, and that this knowledge is not necessarily shared by John. These facts are expressed in axioms ($A24$) and ($A25$).

$$\textbf{True}(\textbf{Know}(\text{Rob}, \text{Tool}(\text{B1}) = \text{WR1})) \qquad (A24)$$

$$\textbf{True}(\textbf{Know}(\text{Rob}, \text{Location}(\text{WR1}) = \text{Location}(\text{TB1}))) \qquad (A25)$$

The axioms for illocutionary acts have been described in Chapter V. Chapter VI discussed axioms for surface speech acts and focusing, while Chapter IV presented a plan involving the action of moving. The new actions of unfastening and removing are straightforward physical actions and will not be presented here. The only deviation from previous examples is the additional condition that both robots and humans can perform illocutionary acts, but only humans can perform physical actions.

The following notation is used for the illustrations in this chapter: Each node in the plan has some sort of boldface label (**P1**, **P2**, etc.) to make it easier to refer to. Dotted boxes are used to represent phantom goals. The successor relation between actions is represented by solid connecting lines, and hierarchical relationships by dotted lines. Each node has an associated world. For goal nodes, the world is written inside parentheses (e.g., (W_i)), to represent that the planner is to start in world W_i and find some actions to reach a world in which the goal is satisfied. For phantom nodes, the world name is not in parentheses to indicate the goal is actually satisfied within the indicated world. Action nodes have a label like "$W_i \rightarrow W_j$" to indicate that the action is a transformation relating worlds W_i and W_j. Actions will often be planned without knowing precisely what worlds they will be performed in, or precisely what world will be the result of the action. This is particularly true of actions that are represented at a high level of abstraction. Worlds are represented in the diagram as "?" if at that point the planner has not yet assigned a definite world. (Note that KAMP can often reason about what is true at a given point in the plan, even though it has not assigned a world to the node, since frame axioms can be stated for high-level actions that describe some changes and leave others unspecified.) A notation like "$W_1 \rightarrow ?$" is assigned to a high-level action that may be expanded to several actions at a lower level. The planner knows the action sequence will begin in W_1 but it will not know the resulting world until the action is expanded. A notation like "$? \rightarrow ?$" is used when the planner knows where in a sequence a high-level action must fall in relation to other actions in the plan, but cannot assign either an initial or final world.

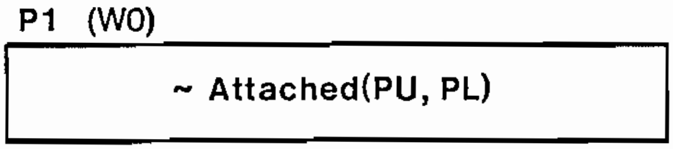

Figure 7.2

The Initial Procedural Network

2. Planning the Utterance

The top-level goal that is given to Rob (and thus to KAMP) is

$$\mathbf{True}(\sim \mathrm{Attached}(\mathrm{PU}, \mathrm{PL})).$$

It is also necessary to tell KAMP which agent is doing the planning. If it knows that Rob is doing the planning, then it can assume that Rob will want to do any action that satisfies a goal, while this condition must be verified explicitly for any agent other than Rob (see Chapter IV).

The first thing KAMP does is create a procedural network from the goal. This initial goal is depicted as node P1 in Figure 7.2. Once the initial procedural network is created, KAMP proceeds as outlined in Chapter IV to expand the initial goal node into a plan. As you will recall, KAMP proceeds in a series of cycles in which each goal node and high-level action is expanded. Then critics examine the plan, making modifications based on the detection of global interactions. The actions in KAMP's domain are divided into three abstraction levels: the high-level actions are the illocutionary acts and the physical action of removing; the next level consists

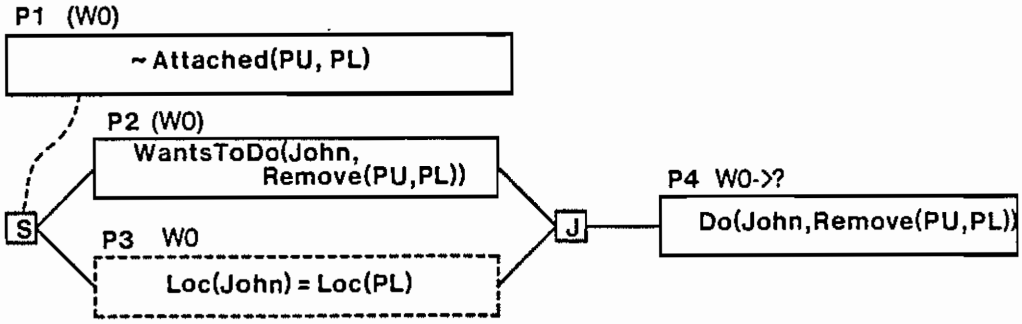

Figure 7.3

Rob Plans for John to Remove the Pump

of surface speech acts and concept activation actions; the lowest level consists of description planning, utterance of sentences, unfastening, getting, and moving. When KAMP has performed enough expansion-criticism cycles so that the entire plan has been fully expanded to the next lowest level of abstraction, it verifies that the plan works by proving that the top-level goal is true in the world resulting from the performance of the actions planned.

Returning to the example, after KAMP has created the initial network, it tries to show that Rob knows the goal is satisfied in the current state of the world, W_0. Since the goal is not satisfied, further planning is required, resulting in the procedural network depicted in Figure 7.3. KAMP consults the action summaries to see if there is any action it knows about at this level of abstraction that achieves the goal as one of its effects. The action of removing has the desired effect, but the action preconditions say that only humans can perform removing actions, and since Rob is not a human, KAMP plans to achieve the goal by having John remove the pump (creating a node, **P4 of** Figure 7.3). To have John remove the pump, KAMP must also establish the preconditions that John *wants* to remove the pump (node **P2**) and that John be in the same place as the pump (node **P3**). Because the latter

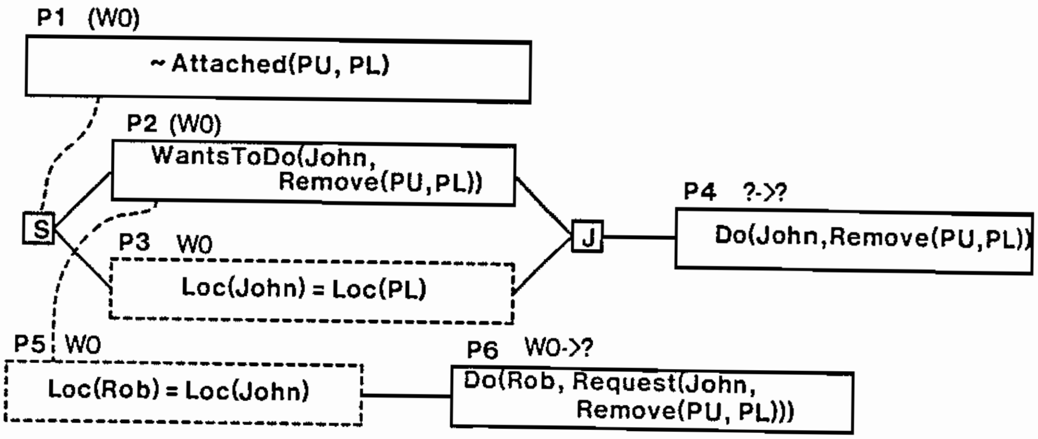

Figure 7.4

Rob Requests that John Remove the Pump

is already satisfied, it is marked as a phantom, and attention focuses on **P2**.

In the next expansion-criticism cycle of the highest abstraction level, KAMP tries to show that John wants to remove the pump in world W_0. Since there is no knowledge to support that conclusion, KAMP follows its procedure of checking action summaries and selecting an action that is likely to achieve the goal. The action summaries indicate that the REQUEST action has the intended effect, so KAMP plans for Rob to request of John to remove the pump. This leads KAMP to construct the procedural net represented in Figure 7.4. At the highest level of abstraction, a complete plan has now been formulated. Therefore, KAMP attempts to prove that the plan it has proposed so far actually works. The verification step succeeds, and KAMP proceeds down to the next level in the abstraction hierarchy.

The next level of abstraction is very important because this is where utterances are introduced into the plan. The first actual linguistic choice KAMP has to make is how to expand the REQUEST action. The expansion procedure for REQUEST

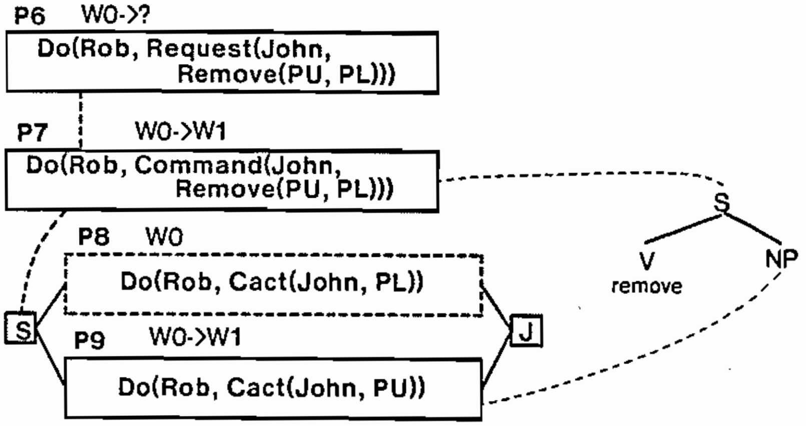

Figure 7.5

A Request Expanded as an Imperative Utterance

has procedurally embedded knowledge that requests can be realized as imperatives. Since syntactic variations in requests are primarily motivated by politeness considerations and KAMP currently does not have an adequate formalism for reasoning about politeness, the imperative is adopted by default. Figure 7.5 shows the expansion of the REQUEST as a COMMAND.

KAMP also makes deductions at this point about what deep case arguments of the predicate are going to be filled in the utterance realizing the REQUEST. The verb has not been chosen at this point — KAMP is still gathering information that will enable it to make that choice. It may be obvious from what is currently in global focus, or what is generally known, that some case arguments can be inferred. For example, if the speaker knows that the hearer knows that the only thing the pump is attached to is the platform, then it is not necessary to say "Remove the pump *from the platform.*" If the pump was attached to several things, but the platform was currently in global focus, and the speaker says, "Remove the pump," and the hearer believes it is consistent with the speaker's intentions to remove it from the platform, then the hearer will assume that the speaker intends the pump to be removed from the platform. Whenever KAMP plans to refer to or describe an action or situation, it checks to see whether the speaker knows that the hearer can make inferences about the case arguments.

In Figure 7.5, KAMP has decided to perform the surface speech act of commanding John to remove the pump, PU, from the platform, PL. There is no way that John can infer PU from his general knowledge, so whatever verb is finally chosen, PU must be mentioned. The situation is different for PL, because KAMP has reasoned that the hearer knows that the pump can only be attached to one thing (using axiom ($A13$)), and that he knows that it is currently attached to the PL (axiom ($A17$)), so it is not necessary to mention PL. KAMP inserts the concept-activation action into the plan, and marks it as a phantom (node **P8**). The phantom action will not necessarily be reflected in the final utterance — it can be noticed by critics and later reactivated if the critic decides that by referring to the platform with an appropriate description it could satisfy another goal.

Once KAMP knows which deep case arguments are mandatory and which are optional, it can select a verb from the lexicon that most adequately matches the case argument requirements. Quite frequently there is only one appropriate verb, so the verb-choice problem does not arise. However, there are a number of instances where several verbs can describe the same event from different perspectives [24], [25]. An often-cited example is that of "buy" and "sell." "Buy" requires explicit mention of the buyer and the object, while "sell" requires explicit mention of the seller and the object. In either situation, the optional case argument can be included as a prepositional phrase (e.g., "John bought a car from Bill," "Bill sold a car to John").

If KAMP determines that one of the case arguments can be eliminated because it can be inferred by the hearer from general knowledge or from global focus, then the verb will be selected that allows the optional argument to be omitted. A partially completed syntax tree is constructed, and nodes in the tree are associated with the COMMAND node, as shown by the dotted lines from the plan to the tree in Figure 7.5.

Since the request is the only illocutionary act that has been planned so far, there is no more linguistic planning to be done at this stage. KAMP now turns its attention to expanding the REMOVE action. Since REMOVE is a physical action, KAMP proceeds exactly as outlined in Chapter IV. Removing something requires removing each of the fasteners that attach it to whatever it is connected to. In this case, it requires removing the bolt B1, since axioms (*A*18) and (*A*19) state that B1 fastens PU to PL. To unfasten a fastener, it is necessary to use some sort of tool appropriate for the particular fastener. At this point the plan is formed using the intensional description, Tool(B1), meaning something like "the tool for removing B1." The action-specific and universal preconditions for unfastening are inserted into the plan, giving the procedural net of Figure 7.6. The precondition nodes are **P10**, **P11** and **P12** — that John knows what the tool is, that John is in the same place as the platform, and that John has the tool.

Since John is already in the same location as PL, the location goal, **P11**, is a phantom. Rob does not know whether John has the tool, nor does Rob even know that John knows what the tool is. Therefore, KAMP plans for Rob to inform John that the tool for removing bolt B1 is wrench WR1, leading to the plan shown in Figure 7.4.

When node **P10** has been expanded, KAMP has reached the point illustrated

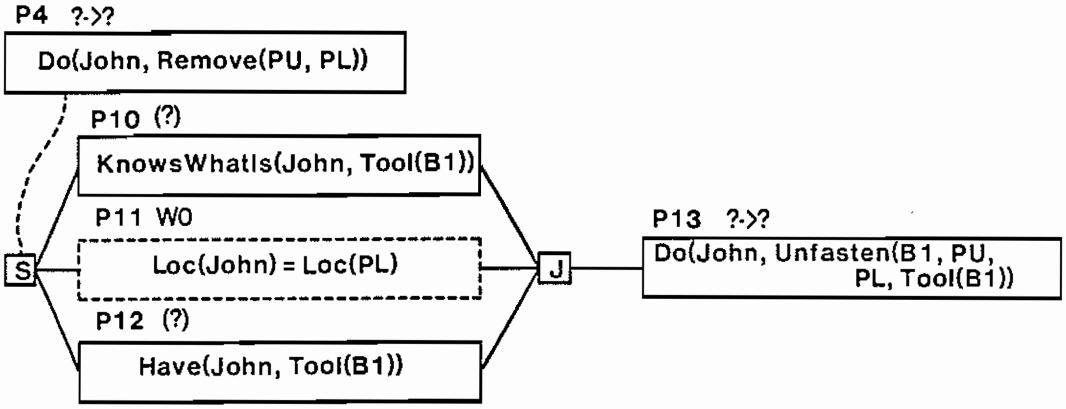

Figure 7.6

KAMP's Plan for Removing the Pump

in Figure 7.7, and the criticism portion of the expansion-criticism cycle begins. As explained in Chapter IV, the critics each have a simple test that they apply to the plan to see if they are applicable. The action-subsumption critic's test works by examining pairs of illocutionary acts such as the newly introduced informing action **P16** and the request, **P6**, to see if they are connected in a way that permits action subsumption, as described in Chapter VI. It uses standard strategies to find connections between the two actions, the most obvious strategy being to examine the explicitly occurring deep case arguments of an event or action predicate referred to one act to see whether the other act comprises an inform of some property of the case argument. Sometimes a deep case is only implicit. For example, almost any physical action can be assisted by some sort of tool, but this tool, or instrument, need not be explicitly present in the underlying predicate. Axioms like ($A22$) and ($A23$) define an implicit instrument case for REMOVE that the action-subsumption critic can take advantage of.

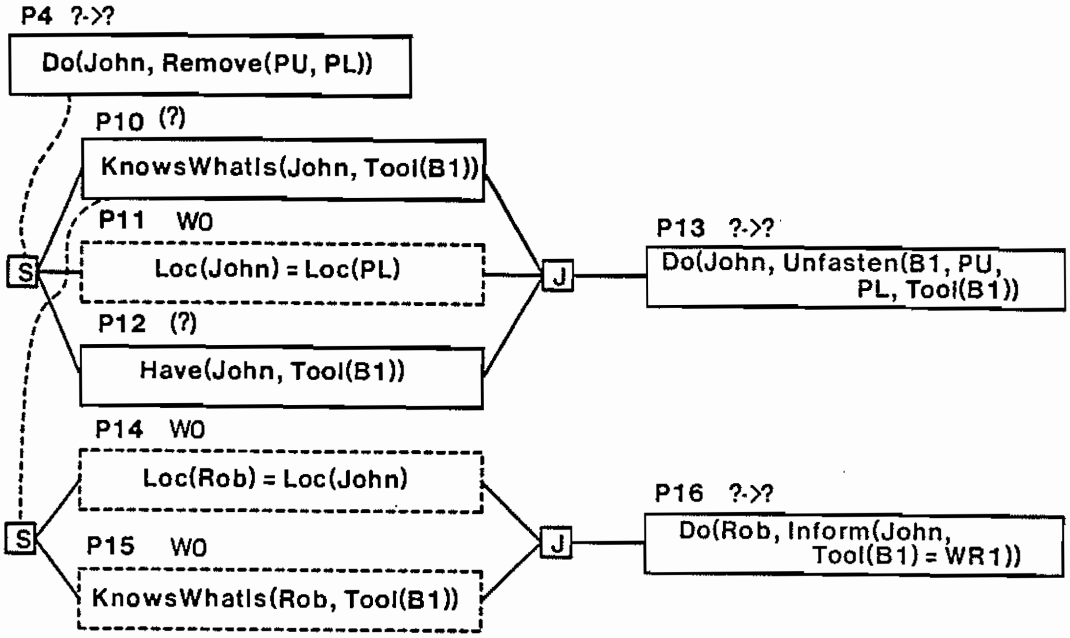

Figure 7.7

John Needs to Know what the Tool Is, so Rob Tells Him

The action-subsumption critic notices that the informing action (**P16**) can be subsumed by the request (**P6** of figure 7.5), provided that reference to the instrument is made explicitly in the utterance. The action-subsumption critic must also determine whether all the preconditions for the subsumption candidate are also satisfied in the state of the world when the subsuming action is going to be "performed." All the conditions, namely that Rob is in the same location as John and Rob knows that Tool(B1) = WR1, are satisfied in this situation, therefore, an action to activate the concept of WR1 is added to the plan as part of the expansion of the COMMAND (**P7**) as node **P18** in Figure 7.8, after checking that such an addition can be accomodated by the choice of verb and syntactic structure for the sentence.

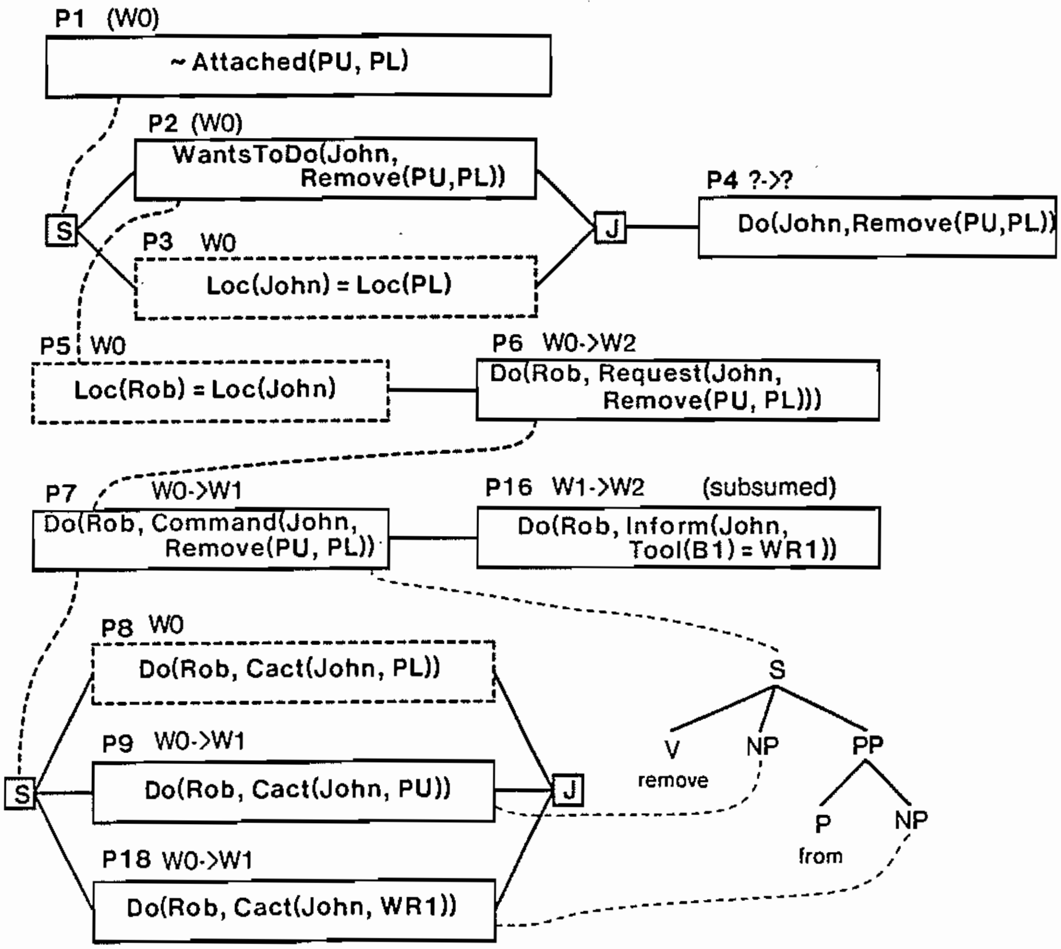

Figure 7.8

Subsuming the Informing Action

Once the critic determines that action-subsumption is the right thing to do, it then moves the INFORM to immediately after the REQUESTand modifies the syntax tree attached to the COMMAND action to include an additional prepositional phrase.

Then the expansion of REMOVE is eliminated and replanned, since the hearer's knowledge has changed because of the informing action being subsumed, and the different knowledge can make a difference in the expansion of the plan. Since it is impossible in general to determine in advance just what this effect will be, that entire portion of the plan is discarded and replanned. Through detailed examination of the discarded actions and the interactions of their effects, it may be possible to avoid totally replanning large portions of the plan, thus saving a great deal of computational effort. This is an example of the type of efficiency considerations that were ignored in the implementaion, partially contributing to the slowness of the system. Figure 7.8 shows the procedural net after criticism by the action-subsumption critic. Note that the REMOVE action has not been expanded, but **P16** remains in the new net as a legacy from the previous expansion.

In this case, the expansion of the informing action is not too much different than the first time, except that the goal of John knowing that WR1 is the right tool for removing the pump has been satisfied by the inform that has been incorporated into the request, and is marked as a phantom. This goal is the analogous goal to **P10** in Figure 7.7, which is referred to as **P10′** in subsequent diagrams such as Figure 7.10. Since both goals analogous to **P10** and **P11** are marked as phantoms, the planner turns its attention toward goal **P12**, that John has WR1 in his posession. For John to have the wrench, he has to know where it is, and he must go there and get it, and this requires that he know where the wrench is. According to our model, John does not have this knowledge, but Rob does (according to axiom ($A25$)), so KAMP plans for Rob to perform an additional informing action to tell John the wrench's location.

The action-subsumption critic now realizes that there is a situation analogous to the one with informing action **P16**. The new informing action (represented as node **P17** in Figure 7.9) is a candidate for subsumption by the request because it informs the hearer of a property of one of the case arguments of the main verb being planned for the request. As in the previous case, the INFORM is relocated so that it follows the REQUEST, and the part of the plan that may be affected by the

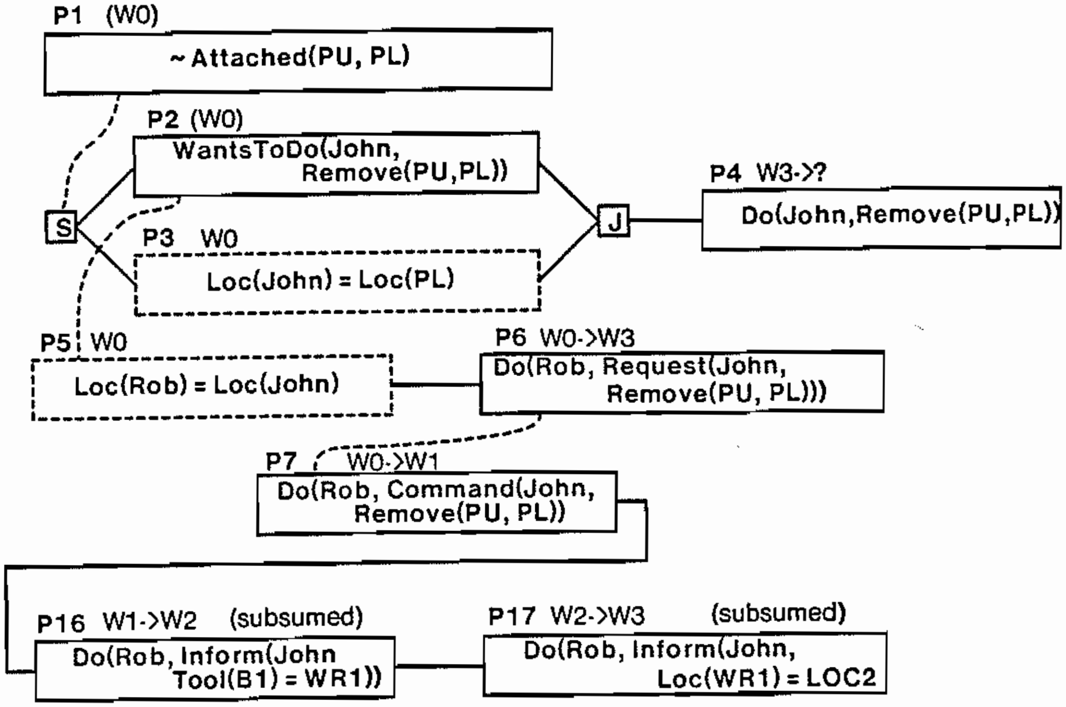

Figure 7.9

The Second Informing Action Subsumed

hearer's new knowledge is discarded and replanned, as before. Figure 7.9 depicts the procedural net after this last round of criticism.

When the REMOVE action is expanded for the third time, the goals involving John's knowledge (**P10**, **P18**, **P21** and **P22** of Figure 7.10) are marked as phantoms. On the next criticism pass, the resolve-conflicts critic will notice that the action of John moving to the tool box to get the wrench undoes the phantom goal that John is at the platform so he can remove the pump. The conflict-resolution critic proposes linearization of the split so the goal of John being at the platform is achieved after he goes to the tool box and gets the wrench. Figure 7.10 shows the plan after the criticism by the conflict-resolution critic and the expansion of the goal of John being at the platform into the MOVE action, **P23**.

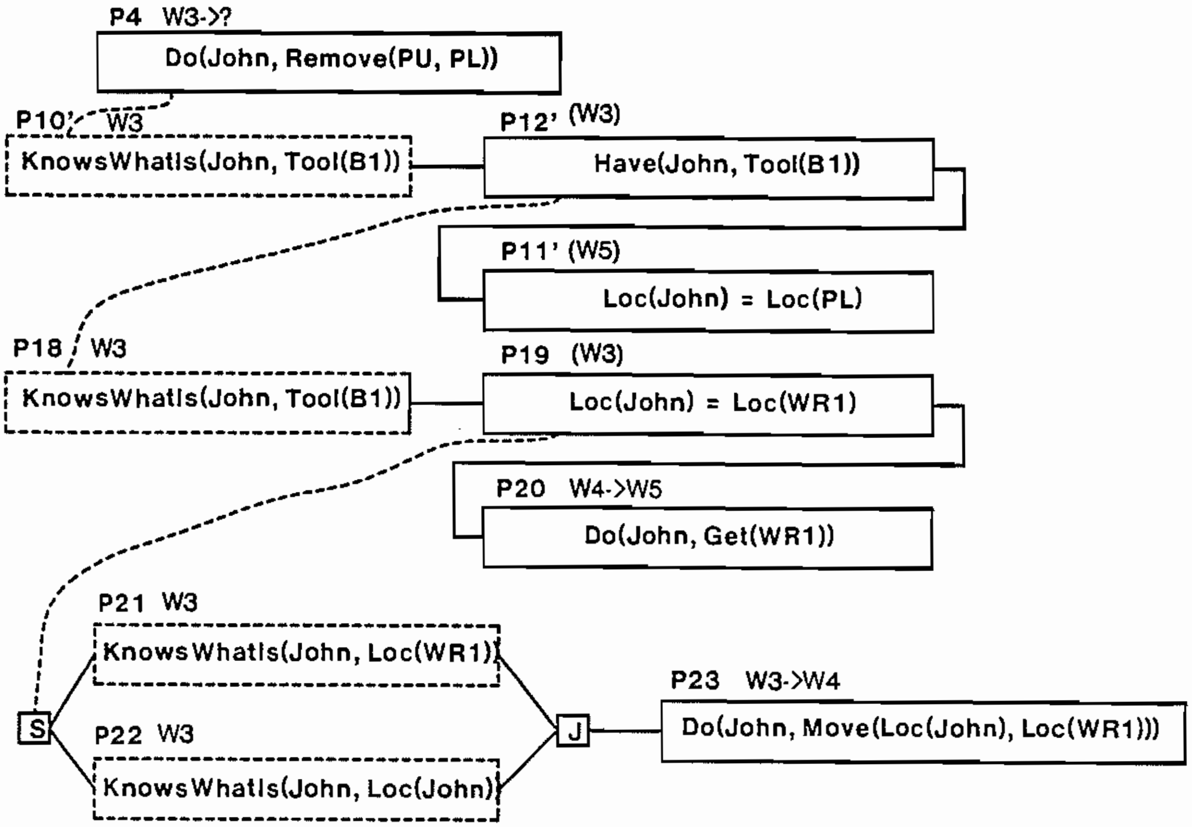

Figure 7.10

After Conflict Resolution

At the point depicted in Figure 7.10, the plan is ready to be expanded down to the final (lowest) level of abstraction, where some specific syntactic choices are made, descriptions for the concept activations are chosen, lexical choices are made, and the final utterance is produced. This is the appropriate time for KAMP to consider the effect of utterances on focus. When KAMP starts expanding actions at this level, it first checks to see what is in focus or potential focus. This knowledge is useful for choosing a syntactic structure for the sentence and for choosing descriptors and generating pronominal references. Since the example is assumed to be the initial utterance in a dialog, there is no immediate focus, however some objects can be in global focus because of general world knowledge and knowledge about the state of the task (see Grosz [32]).

If focusing requirements do not suggest that any particular marked syntactic structure is necessary, then the simplest default structure is chosen for the sentence, and expansion of the concept-activation actions continues. As discussed in Chapter VI, the process of expanding concept-activation actions involves the selection of some mutually believed description of the intended concept. The planner begins by asserting the existence of a typical possible world compatible with the kernel of Rob and John's knowledge. If the concept being activated is mutually believed to be in focus and the pronominalization rules indicate that pronominal reference is possible, then a pronoun of the appropriate number and gender is chosen. It is possible that a pronoun is not chosen, even if it is consistent with the focusing rules, if descriptors have been added to the concept activation action as a result of an action subsumption taking place. Otherwise, descriptor selection begins by choosing a *basic-level* descriptor. Basic-level descriptors, as defined by Rosch [85], are descriptors that describe an object as belonging to a category that is assumed by the speaker and hearer to be the "level of abstraction at which the organism can obtain the most information with the least cognitive effort." For example, "chair" is the basic-level descriptor of objects in an abstraction hierarchy that includes "furniture" as a superordinate and "recliner" as a subordinate. Basic-level information is useful to KAMP not only for planning the head noun of a noun phrase, but also for applying "lexical generalization" strategies to inform the hearer about properties of objects in focus. KAMP knows about basic-level descriptors for different objects in the domain, and when this default predicate is shown to be mutually believed by the speaker and hearer, it is automatically incorporated into the description.

The next step is to assure that the speaker can identify the object from the description provided. KAMP tries to generate a minimal description that serves to distinguish the object from others in focus. The minimal description strategy seems reasonable, and there is some psychological evidence to suggest its validity, (e.g., Olson), but one does not have to examine very many dialogs to find counterexamples to the hypothesis that people always produce minimal descriptions. According to the language generation theory embodied in KAMP, people do generate minimal descriptions for concept activations, but these descriptions can be augmented for a variety of reasons, for example, to realize additional informing actions (as in this example) or to make it easier for a speaker to identify an object when an identification is planned (see Cohen [18]).

KAMP does not produce a provably minimal description, since that would involve solving an NP-complete set covering problem. It simply selects a set of descriptors sufficient to uniquely identify the concept in the current context, without adding any extra ones. When the final utterance is produced, the referring expression will contain descriptors added by the action subsumption critic as well as those necessary for identification of the concept.

Once a set of descriptors for each concept activation is chosen, the descriptors must be realized linguistically. This process may be quite complex, but for KAMP it has been simplified by eliminating some of the intricacies of lexical choice by assuming a straightforward correspondence between the predicates used as descriptors and English words. Therefore, each predicate will have a realization as a noun, adjective, or some realization strategy that involves the planning of a prepositional phrase or relative clause.

In the example here, one of the descriptors chosen (the descriptor of the wrench as in the location of the tool box) involves the use of a prepositional phrase, the object of which will be some description of TB1 (the tool box). The prepositional phrase requires planning another concept activation of TB1, so an appropriate concept-activation node is inserted into the plan, and this node is expanded on the next cycle of the planner.

After the prepositional phrase has been planned, the utterence is close to being in its final form. The concept-activations are linearly ordered to correspond with the order in which the constituents that realize them are ordered in the syntactic structure. This permits the computation of the worlds resulting from the actions. The only necessary modifications are simple syntactic and morphological alterations to ensure subject-verb agreement, and the correct endings on auxiliaries. These processes are automatic and regular and have nothing to do with the speaker's intentions or the hearer's knowledge, so this final step of processing is reserved for a final pass that prints the plan and any utterances that are part of the plan.

The final utterance produced by KAMP is illustrated in Figure 7.11, which shows only language-related parts of the plan. The utterance is, *"Remove the pump with the wrench in the tool box,"* which KAMP has reasoned will realize the request (**P6**) and the informing actions (**P16** and **P17**).

3. Conclusion

This chapter has described how KAMP plans utterances by examining a single example in detail. Of course, this is just one instance of a large class of situations in which KAMP is capable of performing.

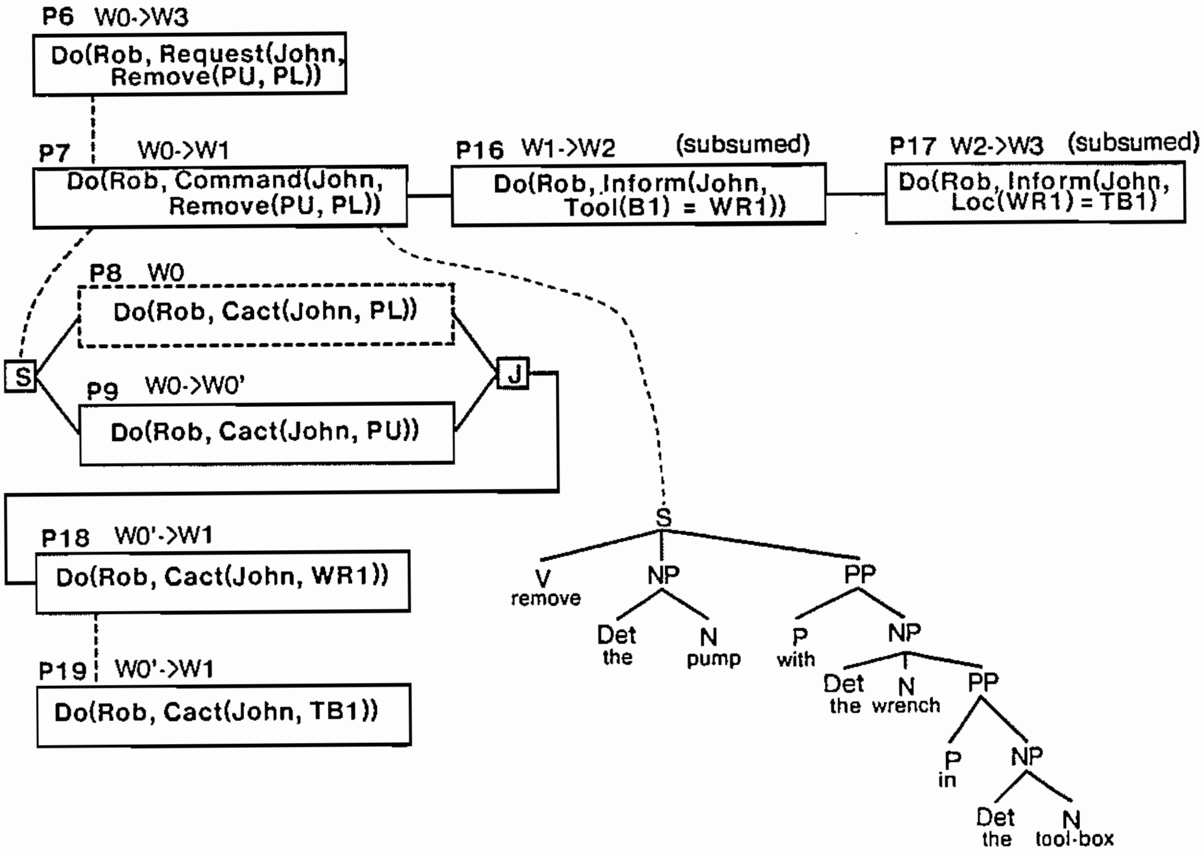

Figure 7.11

The Final Utterance Plan and Syntax Tree

KAMP does not currently have a very large or sophisticated grammar, since most effort has been directed toward bridging the gap between abstractly specified illocutionary acts and surface English sentences. For this reason, most of the problems of lexical choice, representation of grammatical knowledge, and reasoning about social goals have been reserved for future research.

KAMP is designed to perform well in planning illocutionary acts that satisfy a speaker's goals involving the knowledge and wants of other agents. KAMP can then examine the plan containing illocutionary acts and can plan appropriate utterances to realize them using action-subsumption strategies such as adding modifiers to noun phrases to achieve multiple goals in a single utterance. KAMP is also capable of reasoning about how a speaker's physical actions affect the hearer's knowledge of the speaker's intentions, and can plan to use actions such as pointing in conjunction with linguistic actions to achieve the speaker's goals.

The experience gathered during the implementation of KAMP is that the problem-solving techniques described here constitute a feasible approach to producing utterances that satisfy multiple goals. Although planning as a practical approach to natural-language generation has yet to be demonstrated, this research has taken the first step in that direction.

VIII

CONCLUSION

0. What Have We Learned?

The research described in this thesis has focused on the problem of how speakers plan utterances, that satisfy multiple goals. Producing such utterances given only a description of the speaker's goals, is not a simple process; it requires a powerful system that is capable of general reasoning about agents' beliefs and intentions. It is difficult to envision any alternative to language planning that will account for the wide range of behavior.

It has been demonstrated that agents must plan *both* physical and linguistic actions to satisfy their goals, and that physical and linguistic actions interact with one other. For example, an agent may plan to perform some physical action and to carry out the action he needs to have certain knowledge. This leads him to plan a linguistic action such as asking a question. In the course of asking the question, he may need to refer to some object for which the speaker and hearer have no convenient mutually believed description. This may lead to the planning of a physical action of pointing to indicate his intention to refer. This in turn may require the planning of other physical actions, such as moving close to the object to be pointed to, which in turn may require more planning, even to the point of planning another linguistic action. As well as interacting with physical actions, linguistic actions can interact with each other, and a system that can detect interactions can take advantage of them in constructing surface realizations of illocutionary acts that satisfy multiple goals.

Because of the interactions between linguistic and physical actions, a uniform process that plans both kinds of actions as part of the same plan is desirable. It is conceivable that a language-generation system that is not based on planning could produce utterances that satisfy multiple goals, provided that it was given appropriate input. However, only through the union of language generation with a general planning process that the utterances produced can be fully integrated with the speaker's overall plan.

The KAMP system is an important vehicle for the investigation of a theory of language generation based on planning. Planning as motivated by language generation is different from the planning most often studied in the AI literature in that it requires a planner to reason about intensional concepts such as knowing and wanting. Reasoning about such concepts requires a knowledge representation and deduction system of sufficient generality and flexibility to deal with the complex problems that arise. For this reason, the possible-worlds-based representation outlined in Chapter III was chosen as the basis of KAMP's reasoning mechanism.

The adoption of the possible-worlds formalism presents some problems for a planner, since goals are stated with respect to infinite sets of possible worlds. KAMP's two-stage axiomatization of actions, using action summaries as a heuristic guide to forming plans that can be verified within the possible-worlds formalism is a solution to this problem, allowing efficient plan generation while taking advantage of the representational power of the formalism.

Adapting KAMP from a general-purpose hierarchical planner to a language planner involved axiomatizing the various linguistic actions (illocutionary acts, surface speech acts, focusing, and concept activation) in terms of the possible-worlds formalism, integrating procedures for the expansion of high-level actions into KAMP, and designing critics to examine the plan for fortuitous interactions between parts of the plan, enabling KAMP to integrate the actions by applying action-subsumption strategies into a surface utterance that satisfies multiple goals.

The result of incorporating these capabilities into KAMP is a system capable of producing English sentences as part of an agent's plan. Characteristically, the plans that KAMP produces will involve the cooperative actions of at least two agents and involve both physical and linguistic acts. In producing these plans, KAMP draws on knowledge about the physical situation, each agent's knowledge of the situation, and their knowledge about each other's knowledge in addition to the basic axioms about the actions the agents are capable of performing.

The above discussion outlines the major features of KAMP and highlights its strong points. There are a number of problems with KAMP's performance that were beyond the scope of this research to resolve, and which must be left to future research. First of all, KAMP is slow, and a great deal of work must be done to bring the time required to solve a problem into the realm of practicality. Much of this work consists of solving straightforward engineering problems such as improving the efficiency of the underlying theorem prover and ensuring that the planner avoids duplicated effort in re-expanding a node after a critic has proposed re-ordering actions in the plan. Even the underlying LISP system contributed to the problem, with stack fragmentation accounting for many wasted cycles.

Other problems are of a more fundamental nature. Moore [74] noted a problem with the possible-worlds formalism resulting from the expression of knowledge about knowledge as antecedent rules. When an agent knows many facts, much time can be wasted by a deduction system invoking unneeded antecedent rules. This was never a problem in the small examples considered by Moore, but was definitely a major factor in KAMP's performance, particularly in reasoning about what one agent knows about another's knowledge after four or five actions have been performed. Effort needs to be devoted to alternate axiomatizations of the possible-worlds semantics that avoid this problem.

KAMP was mainly intended to address problems involving the interaction of planning and language generation. This means that there is plenty of room for the extension of both KAMP's problem-solving and linguistic capabilities. KAMP's representation of grammatical knowledge, as discussed in Section 1 of Chapter VI needs to be more modular. KAMP's syntactic coverage of English needs to be expanded, particularly to include more complex noun-phrase constructions. KAMP does not currently produce relative clauses, posessives, or complementized noun phrases. It does not generate sentences with quantifiers, and its handling of negation and indefinite reference doesn't cover all the possibilities that exist. The ability to reason about the hearer's recognition of the speaker's intentions has to be extended to the lower levels of linguistic planning, such as lexical choice, since a large number of situations in which human speakers satisfy multiple goals currenly lie outside of KAMP's abilities.

In spite of these shortcomings, KAMP represents significant progress because it has demonstrated that planning is a feasable means of producing natural language utterances.

1. What's Next?

This section discusses some areas of research that may be profitably pursued, given the foundation that has been laid by the research reported in this thesis.

KAMP's design was motivated by the need to plan natural language, but KAMP's usefulness is not limited strictly to applications involving language generation. With additional effort, KAMP could be useful for a variety of applications that involve both planning and reasoning about knowledge. For example, KAMP could reason about acquiring knowledge. Currently it does this only in the context of asking questions to get information, but it could also plan physical actions that result in acquiring knowledge. One application might be to plan laboratory experiments, where the experiment is designed to verify some hypothesis.

Another non-language-oriented application of KAMP would be as part of a general multiple-agent problem-solving system. The current version of KAMP forms plans involving two agents, but the multiple-agent planning problems have been subordinated to the language-planning problems in this research. As a result, the planning problems that have been solved by KAMP have been relatively simple. Research needs to be devoted to problems involving cooperation among more than two agents and situations in which an agent needs to figure out who knows some information he needs, for example, where there is no clearly defined "expert" who is known to know most facts about the domain. Other interesting situations arise when agents are not always mutually aware of each other's actions.

There are a number of more language-oriented problems that appear to be tractable for a planning system like KAMP. One such problem is the planning of extended discourse. Currently, KAMP plans only very simple dialogs. It may plan more than one utterance if it wants to perform several illocutionary acts and it cannot figure out a way to subsume any of them. The resulting dialogs will be coherent because the illocutionary acts are naturally tied together by being part of the same plan. However, to move beyond simple dialogs consisting of alternating requests and informings, more complex, abstract discourse-level actions must be defined. Such actions would have strategies for their expansion into illocutionary acts. For example *instructing* would describe a plan, or *explanation* would describe a causal chain, employing strategies about the best way to explain plans or causal chains. The planner would then determine the best way to apply the general strategy to the specific situation. This research would involve integrating McKeown's work [69] into a planning framework.

KAMP currently keeps track of focus primarily so it can generate appropriate referring expressions. When planning an extended discourse, the planner would also be concerned about the speaker's need to inform the hearer of topic shifts. Topic shifting actions, as described by Reichman [81], must be formalized and planned when appropriate.

The domains in which KAMP has been applied, such as the calendar problem described in Chapter IV, have been somewhat fanciful in that they assume the existence of robots that have many human-like properties. For example, in the calendar problem the robot could move about freely, it had vision, and it could manipulate objects. There are a number of more practical problems that require some of the same capabilities demonstrated by KAMP. For example, a suitably sophisticated terminal can perform pointing actions via some sort of display enhancement, and can "see" a user's pointing actions with a device such as a mouse. This would make it possible for two agents to carry out a natural language conversation in which deictic actions arise naturally, and the domain does not require the assuming the existence of technology that does not already exist.

2. Conclusion

The primary contribution of this thesis is the demonstration of the feasibility of planning as an approach to natural-language generation. It has focused on the interactions between the specification of utterances as illocutionary acts and the production of grammatical sentences. Although much work, both engineering and basic research, needs to be done to apply the ideas presented here to practical systems, this research takes one more step toward the ultimate goal of building a language-generation system, one that will use language with the same fluency as a native speaker. Although this goal may be a long way off, pursuing it promises to contribute to the development of more gracefully interacting computer systems in the not-too-distant future.

BIBLIOGRAPHY

[1] Allen, James, *Recognizing Intention in Dialogue,* Ph.D. thesis, Department of Computer Science, University of Toronto, 1978.

[2] Allen, James and C. R. Perrault, *Participating in Dialogues: Understanding Via Plan Deduction,* Proceedings of the Second National Conference of the Canadian Society for Computational Studies of Intelligence, 1978.

[3] Appelt, Douglas E., *A Planner for Reasoning about Knowledge and Action,* Proceedings of the First Annual National Conference on Artificial Intelligence, 1980.

[4] Appelt, Douglas E., *Problem Solving Applied to Language Generation,* Proceedings of the Eighteenth Annual Meeting of the Association for Computational Linguistics, 1980.

[5] Austin, J., *How to Do Things with Words,* J. O. Urmson (ed.), Oxford University Press, 1962.

[6] Bobrow, Daniel and T. Winograd, *An Overview of KRL, a Knowledge Representation Language,* Cognitive Science, vol. 1, pp. 3–46, 1977.

[7] Brachman, R. J., *An Introduction to KL-ONE,* in Brachman et al. (eds.) **Research in Natural Language Understanding, Annual Report**, pp. 13–46, Cambridge, 1980.

[8] Brown, Gretchen P., *Toward a Computational Theory of Indirect Speech Acts,* MIT Technical Report MIT/LCS/TR-223, 1979.

[9] Bruce, Bertram C., *Belief Systems and Language Understanding,* BBN Technical Report No. 2973, 1975.

[10] Chafe, Wallace L., *Givenness, Contrastiveness, Definiteness, Subjects, Topics and Point of View,* in Li (ed.), **Subject and Topic**, Academic Press, New York, 1976.

[11] Chafe, Wallace L., *The Flow of Thought and the Flow of Language,* in Givón (ed.), **Discourse and Syntax**, Academic Press, New York, 1979.

[12] Clark, Herbert and C. Marshall, *Reference Diaries,* The Second Conference on Theoretical Issues in Natural Language Processing, 1978.

[13] Clark, Herbert, *Responding to Indirect Speech Acts,* Cognitive Psychology, Vol. 11, No. 4, 1979.

[14] Cohen, Benjamin, *Understanding Natural Kinds,* forthcoming Ph.D. dissertation, Stanford University department of philosophy, 1981.

[15] Cohen, Philip, *On Knowing What to Say: Planning Speech Acts,* Technical Report No. 118, University of Toronto, 1978.

[16] Cohen, Philip and C. R. Perrault, *Elements of a Plan Based Theory of Speech Acts,* Cognitive Science, vol. 3, pp. 177–212, 1979.

[17] Cohen, Philip and H. J. Levesque, *Speech Acts and the Recognition of Shared Plans,* Proceedings of the Canadian Society for Computational Studies in Intelligence, 1980

[18] Cohen, Philip, *The Need for Identification as a Planned Action,* Proceedings of the 7th International Joint Conference on Artificial Intelligence, 1981.

[19] Creider, Chet, *On the Explanation of Transformations,* in Givón (ed.), **Syntax and Semantics**, vol. 12, Academic Press, New York, 1979.

[20] Deutsch, Barbara, *Typescripts of Task-Oriented Dialogs,* SRI International Artificial Intelligence Center Technical Note 146.

[21] Donnellan, Keith S., *Reference and Definite Description,* in Schwartz (ed.), **Naming, Necessity and Natural Kinds**, Cornell University Press, Ithaca, N.Y., 1977.

[22] Doyle, John, *A Truth Maintenance System,* MIT AI Laboratory Memo 521, 1979.

[23] Fikes, Richard E., and N. Nilsson, *STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving,* Artificial Intelligence, No. 2, 1971.

[24] Fillmore, C. J., *The Case for Case,* in Bach and Harms (eds.), **Universals in Linguistic Theory**, Holt, Rinehart and Winston, New York, 1968.

[25] Fillmore, C. J.,*The Case for Case Reopened,* in Cole et al. (eds.), **Syntax and Semantics**, vol. 8, Academic Press, New York, 1978.

[26] Friedman, J., *Directed Random Generation of Sentences,* Communications of the ACM, Vol. 12, No. 6, 1969.

[27] Gabriel, Richard P., *An Organization for Programs in Fluid Domains,* Ph.D. dissertation, Stanford University, 1980.

[28] Goldman, Niel, *Computer Generation of Natural Language from a Deep Conceptual Base,* Stanford Artificial Intelligence Laborabory Memo AIM-247, 1974.

[29] Grice, H. P., *Logic and Conversation,* in Davidson (ed.), **The Logic of Grammar**, Dickenson Publishing Co., Encino, California, 1975.

[30] Grimes, Joseph, **The Thread of Discourse**, Janua Linguarum Series, Mouton Press, The Hague, 1975.

[31] Grishman, R., *Response Generation in Question-Answering Systems,* Proceedings of the Seventeenth Annual Meeting of the Association for Computational Linguistics, pp. 99–101, 1979.

[32] Grosz, Barbara J., *Focusing and Description in Natural Language Dialogs,* in Joshi et al. (eds.), **Elements of Discourse Understanding: Proceedings of a Workshop on Computational Aspects of Linguistic Structure and Discourse Setting**, Cambridge University Press, Cambridge, 1980.

[33] Grosz, Barbara J., *Utterance and Objective: Issues in Natural Language Communication,* Proceedings of the Sixth International Joint Conference on Artificial Intelligence, 1979.

[34] Grosz, Barbara and G. Hendrix, *A Computational Perspective on Indefinite Reference,* SRI International AI Center Technical Report No. 181, 1978

[35] von Hahn, W., W. Hoeppner, A. Jameson and W. Wahlster, *The Anatomy of the Natural Language Dialog System HAM-RPM,* in L. Bolc (ed.), **Natural Language Based Computer Systems**, Hanser-MacMillan, Munich, 1980.

[36] Halliday, M., *Language Structure and Language Function,* in Lyons (ed.), **New Horizons in Linguistics**, Penguin, 1970.

[37] Halliday, M., and R. Hasan, **Cohesion in English**, Layman Press, London, 1976.

[38] Halliday, M., **Language as Social Semiotic**, University Park Press, Baltimore Md., 1978.

[39] Hart, Peter and R. Duda, *PROSPECTOR — A Computer-Based Consultation System for Mineral Exploration,* SRI International AI Center Technical Report No. 155.

[40] Hayes-Roth, Barbara and F. Hayes-Roth, *A Cognitive Model of Planning,* Cognitive Science, Vol. 3, No. 4, 1979.

[41] Hendrix, Gary, *Modeling Simultaneous Actions and Concurrent Processes,* Artificial Intelligence, Vol. 4, No. 3, 1973.

[42] Hewitt, Carl, *Description and Theoretical Analysis (Using Schemata) of Planner: a Language for Proving Theorems and Manipulating Models In a Robot,* MIT Artificial Intelligence Laboratory Tecnical Report AI-TR-258, 1972.

[43] Hintikka, J., **Knowledge and Belief**, Cornell University Press, Ithaca, 1962.

[44] Hintikka, J., *Semantics for Propositional Attitudes,* in Linsky (ed.), **Reference and Modality**, Oxford University Press, London, 1971.

[45] Hobbs, Jerry and D. Evans, *Conversation as Planned Behavior,* Cognitive Science, vol. 4, pp. 349–377, 1980.

[46] Hughes, G. E. and M. J. Cresswell, **Introduction to Modal Logic**, Methuen and Co. Ltd, London, 1968.

[47] Kaplan, S. J., *Indirect Responses to Loaded Questions,* Proceedings of the Second Conference on Theoretical Issues in Natural Language Processing, 1978.

[48] Konolige, Kurt, and N. Nilsson, *Planning in a Multiple Agent Environment,* Proceedings of the First Annual National Conference on Artificial Intelligence, 1980.

[49] Konolige, Kurt, *A First-Order Formalization of Knowledge and Action for a Multi-Agent Planning System,* to appear in **Machine Intelligence 10**, 1981.

[50] Kowalski, Robert, *Logic for Problem Solving,* Department of Computational Logic, School of Artificial Intelligence, University of Edinburgh, Memo No. 70, 1974.

[51] Kripke, Saul, *Semantical Analysis of Modal Logic,* Zeitschrift für Mathematische Logik und Grundlagen der Mathematik, vol. 9, pp. 67–96, 1963.

[52] Kripke, Saul, *Semantical Considerations on Modal Logic,* in Linsky (ed.), **Reference and Modality**, Oxford University Press, London, 1971.

[53] Kripke, Saul, *Speaker Reference and Semantic Reference,* in French et al. (eds.), **Contemporary Perspectives in the Philosophy of Language**, University of Minnesota Press, Minneapolis, Minnesota, 1977.

[54] Lakoff, Robin, *The Logic of Politeness; or Minding Your P's and Q's,* in Papers from the Ninth Regional Meeting of the Chicago Linguistic Society, 1973.

[55] Levin, James A., and N. Goldman, *Process Models of Reference in Context,* USC-ISI Technical Report ISI/RR-78-72, 1978.

[56] Levy, David M., *Communicative Goals and Strategies: Between Discourse and Syntax,* in Givón (ed.), **Syntax and Semantics**, vol. 12, Academic Press, New York, 1979.

[57] Levy, David M., *The Archetecture of the Text,* Ph.D. dissertation, Department of Computer Science, Stanford University, 1979.

[58] Linde, Charlotte, and J. Goguen, *The Structure of Planning Discourse,* UCLA Artificial Intelligence Memo No. 9, 1977.

[59] Linde, Charlotte, *Focus of Attention and the Choice of Pronouns in Discourse,* in Givón (ed.), **Syntax and Semantics**, vol. 12, Academic Press, New York, 1979.

[60] Mann, William C., and J. A. Moore, *The Computer as Author — Results and Prospects,* USC-ISI Technical Report ISI-RR-79-82, 1980.

[61] Matthiessen, Christian, *A Grammar and a Lexicon for a Text-Production System,* Proceedings of the Nineteenth Annual Meeting of the Association for Computational Linguistics, 1981.

[62] McCarthy, John and P. Hayes, *Some Philosophical Problems from the Standpoint of Artificial Intelligence,* in Meltzer and Michie (eds.), **Machine Intelligence**, vol 4, Edinburgh University Press, 1969.

[63] McCarthy, John, M. Sato, T. Hayashi, and S. Igarashi, *On the Model Theory of Knowledge,* Stanford Artificial Intelligence Laboratory Memo AIM-312, 1978.

[64] McCarthy, John, *Circumscription,* Artificial Intelligence, vol. 13, 1980.

[65] McDermott, Drew and J. Doyle, *Non-Monotonic Logic I,* Artificial Intelligence, vol. 13, 1980.

[66] McDermott, Drew, *A Temporal Logic for Reasoning about Processes and Plans,* Yale University Department of Computer Science Research Report No. 196, 1981.

[67] McDonald, David D., *Subsequent Reference: Syntactic and Rhetorical Constraints,* Proceedings of the Second Conference on Theoretical Issues in Natural Language Processing, 1978.

[68] McDonald, David D., *Natural Language Production as a Process of Decision Making under Constraint,* Ph.D. dissertation, MIT, 1980.

[69] McKeown, Kathleen R., *Generating Relevant Explanations: Natural Language Responses to Questions about Database Structure,* Proceedings of the First Annual National Conference on Artificial Intelligence, 1980.

[70] Meehan, James, *The Metanovel: Writing Stories by Computer,* Ph.D. dissertation, Yale University, 1976.

[71] Montague, R., *Syntactical Treatments of Modality with Corollaries on Reflexion Principles and Finite Axiomatizability,* Acta Philosophica Fennica, vol. 16, 1963.

[72] Moore, James A., and W. C. Mann, *A Snapshot of KDS, a Knowledge Delivery System*, Proceedings of the Eighteenth Annual Meeting of the Association for Computational Linguistics.

[73] Moore, Robert C. and G. Hendrix, *Computational Models of Beliefs and the Semantics of Belief-Sentences,* SRI International AI Center Technical Report No. 187, 1979.

[74] Moore, Robert C., *Reasoning about Knowledge and Action,* SRI International AI Center Technical Report No. 191, 1980.

[75] Moore, Robert C., *Problems in Logical Form*, Proceedings of the 19th Annual Meeting of the Association for Computational Linguistics, 1981.

[76] Morgan, Jerry L., *Some Remarks on the Nature of Sentences,* Papers from the Parasession on Functionalism, Chicago Linguistics Society, 1975.

[77] Morgan, Jerry L., *Two Types of Convention in Indirect Speech Acts*, in P. Cole (ed.), **Syntax and Semantics Volume 9: Pragmatics** , Academic Press, New York, 1978.

[78] Nilsson, Nils J., **Principles of Artificial Intelligence**, Tioga Publishing Co., Palo Alto, 1980.

[79] Perrault, C. Raymond and P. Cohen, *Inaccurate Reference,* in Joshi et al. (eds.) **Formalizing Discourse**, Cambridge University Press, 1980.

[80] Reddy, Michael J., *The Conduit Metaphor — A Case of Frame Conflict in Our Language about Language,* in Ortony (ed.), Metaphor and Thought, Cambridge University Press, Cambridge, 1979.

[81] Reichman, Rachel, *Conversational Coherency,* Cognitive Science, Vol. 2, No. 4, 1978.

[82] Reiter, Raymond, *A Logic for Default Reasoning,* Artificial Intelligence, vol. 13, 1980.

[83] Robinson, A., D. Appelt, B. Grosz, G. Hendrix and J. Robinson, *Interpreting Natural-Language Utterances in Dialogs about Tasks,* to appear in Communications of the ACM, 1981.

[84] Rogers, R., **Mathematical Logic and Formalized Theories**, North-Holland, Amsterdam, 1971.

[85] Rosch, Eleanor, C. Mervis, W. Gray, D. Johnson, and P. Boyes-Braem, *Basic Objects in Natural Categories,* Cognitive Psychology, vol. 8, pp. 382–439, 1976.

[86] Sacerdoti, Earl, **A Structure for Plans and Behavior**, Elsevier North-Holland, Inc., Amsterdam, 1977.

[87] Schank, Roger, N. Goldman, C. Reiger and C. Riesbeck, **Conceptual Information Processing**, North-Holland, Amsterdam, 1975.

[88] Schank, Roger and R. Abelson, **Scripts Plans, Goals and Understanding**, Lawrence Erlbaum Associates, New Jersey, 1977.

[89] Scott, Carlisle, W. Clancey, R. Davis and E. Shortliffe, *Explanation Capabilities of Production-Based Consultation Systems,* American Journal of Computational Linguistics, Microfiche 62, 1977.

[90] Searle, John, **Speech Acts: An Essay in the Philosophy of Language**, Cambridge University Press, Cambridge, 1969.

[91] Searle, John, *Indirect Speech Acts,* in Expression and Meaning: Studies in the Theory of Speech Acts, Cambridge University Press, Cambridge, 1979.

[92] Searle, John, *A Taxonomy of Illocutionary Acts,* in **Expression and Meaning: Studies in the Theory of Speech Acts**, Cambridge University Press, Cambridge, 1979.

[93] Sidner, Candace L., *Toward a Computational Theory of Definite Anaphora Comprehension in English,* MIT Technical Report AI-TR-537, 1979.

[94] Simmons, R. F. and J. Slocum, *Generating English Discourse from Semantic Nets,* Communications of the ACM, Vol 15, No. 10, 1972.

[95] Simmons, R.F., *Semantic Networks: Their Computation and Use for Understanding English Sentences,* in Schank and Colby (eds.), **Computer Models of Thought and Language**, W. H. Freeman and Co., San Francisco, 1973.

[96] Stefik, Mark, *Planning with Constraints,* Stanford Computer Science Technical Report STAN-CS-80-784, 1980.

[97] Strawson, P. F., *Intention and Convention in Speech Acts,* in Searle (ed.) **The Philosophy of Language**, Oxford University Press, 1971.

[98] Swartout, William. R., *Producing Explanations and Justifications of Expert Consulting Programs,* MIT Computer Science Technical Report MIT-LCS-TR-251, 1981.

[99] Wahlster, Wolfgang, *Implementing Fuzziness in Dialogue Systems,* in Rieger (ed.), **Empirical Semantics**, Brockmeyer, Bochum, 1980.

[100] Waldinger, Richard, *Unsolved Problems in the Blocks World,* Unpublished talk given at the Workshop on Formal Methods in AI, Monterey Dunes, August, 1980.

[101] Weyrauch, Richard W., *Prolegomena to a Theory of Mechanized Formal Reasoning,* Artificial Intelligence, vol. 13, 1980.

[102] Wilensky, Robert, *Understanding Goal-Based Stories,* Yale University Research Report No. 140, 1978.

[103] Wilensky, Robert, *Metaplanning: Representing and Using Knowledge about Planning in Problem Solving and Natural Language Understanding,* University of California at Berkeley Electronic Research Laboratory Memorandum UCB-ERL M80-33, 1980.

[104] Winograd, Terry, **Understanding Natural Language**, Academic Press, New York, 1972.

[105] Winograd, Terry, *A Framework for Understanding Discourse,* in Just and Carpenter (eds.), **Cognitive Processes in Comprehension**, Lawrence Erlbaum associates, New Jersey, 1977.

[106] Winograd, Terry, *What Does It Mean to Understand Language?* Cognitive Science vol. 4, pp 209–241, 1980.

[107] Wong, H. K. T., *Generating English Sentences from Semantic Structures,* University of Toronto Technical Report No. 84, 1975.