Intentions in Communication

P. Cohen, J. Morgan + M. Pollack (eds)
MIT Press (1990)

Chapter 17

Discourse Processing and Commonsense Plans

Diane J. Litman and James F. Allen

1 Introduction

Work on discourse understanding in artificial intelligence has produced a wide range of seemingly incomparable techniques, each addressing a slightly different problem. A recent paper by Grosz and Sidner (1986) is a first attempt at synthesizing and extending the different approaches into a coherent theory of discourse. Key to their framework is the incorporation of two distinct nonlinguistic components—the attentional (related to focus of attention) and the intentional (related to plans and goals)—into a theory of discourse structure. However, in *task-oriented dialogues*[1] an important distinction must also be made between discourse intentions and another nonlinguistic notion relevant to discourse theory: *commonsense*[2] task knowledge. This distinction is needed to account for subdialogues that do not directly correspond to commonsense tasks, for example, clarification and correction subdialogues. In this paper we present a plan recognition model for task-oriented dialogue understanding that explicitly separates what one explicitly says in a discourse about a task from commonsense knowledge about a task itself. We define a set of commonsense *discourse plans,* plans that can be dynamically generated from execution or discussion of commonsense task plans, and distinguish these discourse plans from the task plans. Furthermore, we distinguish discourse intentions from both task and discourse plans, that is, from commonsense plans to which discourse intentions refer. By incorporating knowledge about discourse plans into a commonsense framework, we can account for a wide variety of subdialogues in a discourse while maintaining the computational advantages of the plan-based approach.

Consider Dialogue Fragments 1–3, portions of a task-oriented cooking dialogue collected at AT&T Bell Laboratories:

Dialogue Fragment 1: Subtask Subdialogues

Teacher: Anyway um, and blend that for about twenty seconds. OK?

This research was supported in part by the Office of Naval Research under contract number N00014-80-C-0197 and in part by NSF grant IST-8504726.

Student: Uh huh.
Teacher: And now add, uh three quarters of a cup of heavy cream.
Student: OK.
Teacher: OK and you are to blend that for at least thirty seconds.

Dialogue Fragment 2: A Clarification Subdialogue

Teacher: OK the next thing you do is add one egg to the blender, to the shrimp in the blender.
Student: *The whole egg?*
Teacher: *Yeah, the whole egg. Not the shells.*
Student: Gotcha. Done.

Dialogue Fragment 3: A Correction Subdialogue

Teacher: OK, and now, um just wash off the parsley and um, put it on a paper towel to drain, and then we'll set up the plates. Are you hungry?
Student: Of course.
Now there looks like there's mud in with this stuff.
Teacher: *Really?*
Student: *Really.*
Teacher: *Oh good, great, well, there's mud in the parsley? We'll skip the parsley.*
Student: *Sure looks like it to me.*
Teacher: *Oh goodness. OK, so*
Student: *I don't do have that much experience.*
Teacher: *OK.*
Student: *With parsley.*
Teacher: OK, take out five plates, 'cause I'm afraid the fish is going to start to get cold.

Fragment 1 illustrates subdialogues that directly correspond to cooking subtasks, and Fragments 2–3 illustrate subdialogues that do not correspond to such subtasks. For the purposes of this paper, we will call task-oriented subdialogues that do not correspond to execution or description of domain tasks *generated* subdialogues. For example, the clarification in Fragment 2 is generated by an underspecified description of a subtask, and the correction of Fragment 3 is generated by a problem encountered during the execution of a subtask. Although previous plan-based approaches (Allen and Perrault 1980; Carberry 1985; Grosz 1977; Sidner 1985) work well for subtask subdialogues (as in Fragment 1), they encounter difficulty in accounting for generated subdialogues such as clarifications and corrections. One reason for this difficulty is that previous plan-based models have

not clearly distinguished between what one says in a discourse about a task and the actual knowledge about the domain task itself. For example, previous theories have only allowed discourses to describe steps in task plans (for instance, Grosz 1977) or to be steps in task plans (for instance, Allen and Perrault 1980). As Fragments 2 and 3 illustrate, there are also many ways that discourses can indirectly refer to task plans, for example, via clarifications and corrections. Although work in discourse structure (Grosz and Sidner 1986; Reichman-Adar 1984) has been more careful about maintaining a distinction between discourse intentions and commonsense task plans, such models have not been concerned with the recognition of task plans. Furthermore, they too have neglected the class of "interrupting" discourse intentions corresponding to generated task-related subdialogues.

In this paper we present a plan recognition model for task-oriented dialogue understanding that explicitly separates discourse intentions from commonsense plans that are the objects of such intentions. We define two types of commonsense plans: (1) *domain plans,* plans used to model the tasks, and (2) *discourse plans,* domain-independent plans that can be generated from execution or discussion of domain plans. Our discourse plans represent a new source of commonsense knowledge, mediating between task plans and an important class of discourse intentions. Our approach has several important advantages. First, the incorporation of discourse plans into a theory of task plan recognition allows us to account for a wider variety of task-oriented dialogues (for instance, dialogues with dynamically generated subdialogues such as clarifications and corrections) than previously considered. Second, the discourse plan formulation provides a clean computational model. Plans are a well-defined method of representation, but more importantly, the techniques of plan recognition can be adapted to identify appropriate discourse plans from utterances. Also, plans that usually are considered discourse plans may themselves become the topic of conversation and thus temporarily be viewed as domain plans. For example, clarifications may themselves become the objects of other clarifications. Since both the task and the generated discourse interaction are represented as plans, we can use the same techniques to handle clarifications to an arbitrary depth of nesting (in principle).

The next sections detail our new model of plan recognition. First, we elaborate on the distinction between domain plans, discourse plans, and discourse intentions. We then outline a simple theory of plans that is adequate for the purposes of this paper and show how discourse intentions, discourse plans, and domain plans are defined in this framework. Finally, we present a plan recognition technique for recognizing and relating commonsense structures of discourse and domain plans and illustrate our model with a detailed example.

2 Domain Plans, Discourse Plans, and Discourse Intentions

In this section we present examples illustrating the distinctions between discourse intentions and commonsense plans, and between commonsense domain plans and commonsense discourse plans. As discussed above, such distinctions are necessary to adequately account for subdialogues that are generated during execution and discussion of commonsense tasks.

Consider Dialogue 4 and figure 17.1, a simple task-oriented dialogue fragment and the commonsense task plan underlying it:

Dialogue 4

(1) Person: I'd like to buy a ticket to New York.
(2) Here's five dollars.
(3) Clerk: The next train leaves in five minutes—better hurry. (handing Person a ticket)
(4) Person: OK, thanks.

Figure 17.1 shows the recursive decomposition of a plan into its subplans, with an implicit temporal ordering from left to right. Thus, TAKE-TRIP consists of a BUY-TICKET plan (which itself consists of three subplans), followed by a GOTO-TRAIN plan, followed by a GETON-TRAIN plan. To simplify the current discussion, we will assume that this TAKE-TRIP plan is prebuilt. Formal definitions of plans, and their construction from observed actions, will be discussed later in the paper.

Let us examine why discourse intentions should not be conflated with commonsense plans. First, discourse intentions are not identical to commonsense plans. The discourse intentions underlying the beginning of Dialogue 4 are roughly (1) that Person intends Clerk to believe that Person would like to buy a ticket to New York (and maybe that Person intends Clerk to believe that (1) was a request for a ticket), and (2) that Person intends Clerk to believe that Person is giving Clerk five dollars to buy the ticket. In contrast, the commonsense knowledge underlying Dialogue 4 corresponds to the task plans to which these discourse intentions refer,

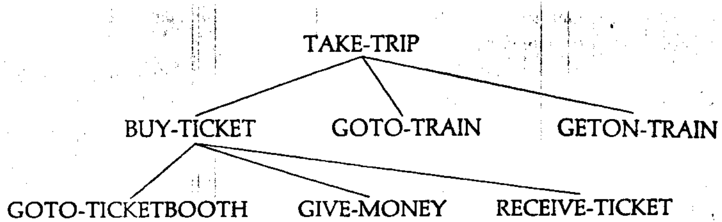

Figure 17.1
Sketch of a commonsense task plan to take a trip.

such as BUY-TICKET and GIVE-MONEY. In other words, commonsense plans are purely abstract structures, whereas discourse intentions are states of mind describable in terms of these structures.[3]

Second, there is no one-to-one correspondence between discourse intentions and steps in the associated commonsense plan. For example, discourse intentions may have only indirect relevance to commonsense plans (for instance, "Thanks" of utterance (4)). Furthermore, besides containing plans referred to in the dialogue (BUY-TICKET and GIVE-MONEY), the task plan contains plans that are inferred using commonsense knowledge, such as TAKE-TRIP, GETON-TRAIN. Although such plans are not explicitly mentioned, their recognition is necessary to provide cooperative behavior such as helpful responses (Allen and Perrault 1980). The helpful response (3) depends on Clerk's inference that BUY-TICKET is a subplan of a larger plan to take a trip. Inferred plans are also needed to account for actions that are never mentioned (Grosz and Sidner 1986), such as Clerk handing Person a ticket (RECEIVE-TICKET).

Third, the structure of discourse intentions does not necessarily follow the structure of the underlying commonsense task plan. Consider Dialogue 5, taken from a corpus of transcripts recorded at an information booth in the Toronto train station (Horrigan 1977) (Dialogue 6 will also be from this corpus):

Dialogue 5

(5) Clerk: Yes.
(6) Person: Can you tell me where to catch the GO trains?
(7) Clerk: Just go down to the bottom of the stairs here.
(8) Person: Do you get tickets down there too?
(9) Clerk: Yes.
(10) Person: Thanks.

Although the discourse intentions involve steps in the task plan of figure 17.1—namely, GOTO-TRAIN and BUY-TICKET—the discourse intentions are presented in a different order than the steps would be executed. This situation is often true of the "information-seeking" form of task dialogues (Carberry 1985).

To summarize, then, discourse intentions (1) are a different sort of entity than commonsense plans; (2) do not necessarily refer to any or every step in the associated commonsense plan; and (3) do not necessarily mirror the execution structure of associated commonsense task plans.

Now let us turn to the claim that in addition to commonsense domain plans, we need to introduce a set of commonsense plans called discourse plans. First, note that discourse intentions may refer to plans generated by, but not explicitly in, the domain plan (for instance, clarifications and corrections; see Litman and Allen 1987). That is, "in addition to the intentions arising from steps in the plan, the intentional structure typically contains... intentions generated by the particular execution of the task and the dialogue" (Grosz and Sidner 1986, 187). Recall Fragments 2 and 3, in which the underlying cooking plan was at times referred to only indirectly, via plans to clarify and correct it. A more complicated example involving the generation of plans from domain as well as from other generated plans is illustrated in Dialogue 6:

Dialogue 6

(11)	Person:	Where do I get the CAN-AM train?
(12)	Clerk:	It's not until six.
(13)		They're meeting at cubicle C.
(14)	Person:	Where's that?
(15)	Clerk:	Just go down here.
(16)		It's on the right-hand side.
(17)	Person:	OK, thank you.

The discourse intention of (11), that Person intends that Clerk intends that Clerk clarify Person's plan to take a trip, is generated when Person realizes that GOTO-TRAIN (again using figure 17.1) cannot be executed, since the location of the train is unknown. The discourse intention of (14) is generated when Clerk's response (13) still does not provide enough information to execute the plan. In other words, these discourse intentions refer to a discourse plan clarifying the domain task and to a discourse plan clarifying this clarification, respectively. Thus, in the case of generated subdialogues, a set of discourse plans such as clarifications and corrections is needed to mediate between the discourse intentions and the domain plans.

Second, discourse plans should be considered part of commonsense knowledge for two reasons. First, discourse plans adhere to the three discourse intention/commonsense plan distinctions discussed above. That is, discourse plans are abstract structures that may be referred to by discourse intentions (as in utterance (11)), that may account for actions that are never mentioned[4] (imagine replacing the linguistic clarifications of utterances (12)–(13) with a nonlinguistic clarification such as handing the person a schedule), and so on. Second, by treating discourse plans and domain plans uniformly, discourse plans may themselves temporarily be viewed as domain plans (as in utterance (14), where a clarification is the domain plan of another clarification), to an arbitrary depth of nesting.

In the next section we present a much fuller discussion of discourse plans. We characterize the set needed, show how to integrate the representation of discourse plans into a domain planning framework, then specify the interaction between domain plan, discourse plan, and discourse intention recognition.

3 *Identifying and Representing Plans*

3.1 *Representing Domain Plans*

Domain plans represent typical tasks that might be performed in a given domain and correspond to the topics of task-oriented dialogues. Such commonsense knowledge has been the mainstay of previous plan-based works. Domain plans are represented as actions and states connected by causality and subpart relationships. Every plan has a *header*, a parameterized action description that names the plan. The *parameters of a plan* are the parameters in the header. As in other planning models (such as STRIPS (Fikes and Nilsson 1971) and NOAH (Sacerdoti 1977)), plans may contain *prerequisites*, *effects*, and a *decomposition*. Prerequisites are conditions that need to hold (or be made to hold) before the plan can actually be applied. Effects are statements that hold after the plan has been successfully executed. We will ignore most prerequisites and effects throughout this paper, except when needed in examples. Decompositions enable hierarchical planning. Although a plan may be usefully thought of as a single action at the level of description of the header, the plan may be decomposed into primitive (that is, executable) actions and other abstract action descriptions (that is, other plans). Such decompositions may be sequences of actions, sequences of subgoals to be achieved, or a mixture of both. For the purposes of this paper we are ignoring all temporal complexities in plans; plans are simply a linear sequence of actions.

Also associated with each plan is a set of applicability conditions called *constraints*. These are similar to prerequisites, except that the planner never attempts to achieve a constraint if it is false. Thus, any action whose constraints are not satisfied in some context will not be applicable in that context.

A library of *plan schemas* will be used to represent knowledge about typical speaker tasks. *Plan instantiations* are formed from such general schemas by giving values to the schema parameters. We will use the term *plan* to refer to both plan schemas and instantiations (the intended meaning should always be clear from the context). A plan recognizer will use the plan schemas to recognize the plan instantiation that produced an executed action. In particular, the recognizer will be concerned with recognizing plans from linguistic actions executed as part of a dialogue. The plan recognizer can build up structures of such instantiations, such as the structure sketched in figure 17.1. Before the process of plan recognition, such a structure is only implicit (and one of many possibilities) in the knowledge base.

Figure 17.2 presents some examples of plan schemas. The first plan indicates a primitive action, GOTO, and its effect, and the second through fourth plans indicate nonprimitive actions. For example, the third

HEADER:	GOTO(agent, location, time)
EFFECT:	AT(agent, location, time)
HEADER:	MEET(agent, arriveTrain)
DECOMPOSITION:	GOTO(agent, gate(arriveTrain), time(arriveTrain))
HEADER:	BOARD(agent, departTrain)
DECOMPOSITION:	GOTO(agent, gate(departTrain), time(departTrain)) GETON(agent, departTrain)
HEADER:	TAKE-TRAIN-TRIP(agent, departTrain, destination)
DECOMPOSITION:	SELECT-TRAIN(agent, departTrain, departTrainSet) BUY-TICKET(agent, clerk, ticket) BOARD(agent, departTrain)
CONSTRAINTS:	EQUAL(destination, station(departTrain)) EQUAL(destination, station(departTrainSet)) EQUAL(departTrain, object(ticket))

Figure 17.2
Some plan schemas for the train domain.

plan summarizes a simple plan schema with header "BOARD (agent, departTrain)," with parameters "agent" and "departTrain," and with a decomposition consisting of the two steps indicated. The prerequisites, effects, and constraints are not shown. Other plans needed in this domain include plans to select trains, plans to buy tickets, and so on. To process dialogues in other domains (such as cooking, as in Fragments 1–3), a different set of domain plans would be relevant.

Throughout this paper we assume that formulas are expressed in a typed logic that will be reflected in the naming of variables. Thus, in the above formulas, *agent* is restricted to entities capable of agency, *departTrain* is restricted to trains, and so on. These types are organized into type hierarchies as commonly found in semantic network formalisms.

3.2 *Discourse Plans*

In addition to domain-dependent task plans, we introduce into the plan-based framework a small set of plans called *discourse plans.* Discourse plans are domain-independent plans that can be generated by other plans in the course of any task-oriented dialogue. For example, discourse plans can be

generated from problems with the execution of a particular task (as in a task correction) as well as from problems with the discourse itself (as in a clarification). Discourse plans allow the recognition of an important class of subdialogues. Like domain plans, discourse plans can be the object of discourse intentions. However, whenever a discourse plan is recognized, recognition of the generating discourse and/or domain plan will also be generated by this recognition.

Our set of discourse plans can be abstractly characterized as the set of ways in which a single plan structure can be manipulated. For the purposes of plan recognition, it is useful to roughly divide the set into three classes: the plans that effectively continue a plan as expected; the plans that involve clarification and correction of a plan; and the plans that introduce new plans. These classes of plans are summarized informally below, via examples:

The Continue Class

TRACK-PLAN: In cases where an executing task action is nonlinguistic, a discourse plan can be generated that involves talking about the execution of this action. An example of this is utterance (2), which describes the handing of the money to Clerk.

The Clarification Class

IDENTIFY-PARAMETER: This discourse plan is generated when a task cannot be executed due to lack of information. In particular, IDENTIFY-PARAMETER provides a suitable description for a term instantiating a parameter of an action such that the hearer is then able to execute the action. For example, to execute the action GIVE-MONEY (P, AMT), P must know some description of AMT stated in terms of dollars, say VALUE-IN-DOLLARS (AMT, 5). Examples of requests for this discourse act include utterances (6), (8), (11), and (14). Examples of the discourse act itself include utterances (7), (9), (12), (13), (15), and (16).

CORRECT-PLAN: This involves specifying correction of a plan in order to ensure its success after unexpected events at runtime. There is no instance of this in the train dialogues, but an example might be to replace utterance (3) with "The fares just went up—you'll have to give me a ten." Fragment 3 also provides an example.

The Topic Shift Class

INTRODUCE-PLAN: This introduces a new plan for discussion that is not part of the previous topic of conversation. Utterance (1) (taken literally) is an example. Usually, however, plan introduction is performed implicitly. For example, utterance (1) only introduces beliefs about a plan into a conversation; it does not reflect speaker or hearer intentions about the plan's execution. In contrast, consider an utterance such as "Give me a ticket to New York." Here, introduction of the plan as a topic is implicit in the request for its execution.

MODIFY-PLAN: This introduces a new plan by performing some modification on a previous plan. There is no instance of this in the dialogues presented above, but an example might be to replace utterances (8)–(9) with "How about the buses?"

Though other discourse plans in each of these classes could be developed, the small set shown is sufficient to understand an interesting class of subdialogues in several domains (Litman and Allen 1987).

Except for the fact that discourse plans have other plans as parameters (and are thus technically *metaplans*),[5] our representation of discourse plans is identical to that for domain plans. In order to define them fully, however, let us introduce some language for referring to other plans. Developing a fully adequate formal theory of this is a large research effort in its own right and cannot be addressed here. Our approach so far is meant to be suggestive of what is needed and is specific enough for our preliminary implementation.

To talk about the structure of plans, we extend our ontology to include plans as objects. We then introduce several predicates that concern plans:

PARAMETER (P, plan): asserts that the term P is a parameter of the action description in the header of the specified plan. More formally, *P* is an instantiation of one of the parameters of the plan schema that *plan* itself is an instantiation of. Given the definition of BOARD in figure 17.2, PARAMETER (S, BOARD (S,TR1)) and PARAMETER (TR1, BOARD (S, TR1)) are true.

STEP (subplan, plan): asserts that *subplan* is part of the decomposition of *plan*. More formally, *subplan* is an instantiation of one part of the decomposition of the plan schema that *plan* itself is an instantiation of. Give the definition of BOARD in figure 17.2 STEP (GOTO (S, gate (TR1), time (TR1)), BOARD (S, TR1)) and STEP (GETON (S, TR1), BOARD (S, TR1)) are true.

Plans are not the only objects whose structure we need to examine. In addition, we will need to refer to parameters of actions and propositions. Thus, we will be working in a logic admitting plans, actions, and propositions as objects. The PARAMETER predicate will be used to make assertions about the structure of all these types of objects.

Figure 17.3 presents the formal representation of two example discourse plans, namely, IDENTIFY-PARAMETER from the clarification class and INTRODUCE-PLAN from the topic switch class. Unlike domain plans, such discourse plan schemas remain constant across all domains.

IDENTIFY-PARAMETER provides a suitable description of a parameter that enables the hearer to then execute an action in the clarified plan.[6] It is

HEADER:	IDENTIFY-PARAMETER(speaker, hearer, parameter, action, plan)
DECOMPOSITION:	INFORMREF(speaker, hearer, parameter, proposition)
EFFECT:	KNOW-PARAMETER(hearer, parameter, action, plan)
CONSTRAINTS:	PARAMETER(parameter, action) STEP(action, plan) PARAMETER(parameter, proposition)
HEADER:	INTRODUCE-PLAN(speaker, hearer, action, plan)
DECOMPOSITION:	INFORM(speaker, hearer, WANT(speaker, action))
EFFECT:	BEL(hearer, WANT(speaker, plan))
CONSTRAINT:	STEP(action, plan)

Figure 17.3
Some discourse plan schemas.

performed by describing the parameter via some proposition. (INFORMREF will be further explained in the section on speech acts.) There are several constraints on the relationship between the discourse plan and *plan* (the plan generating the discourse plan), namely, that *parameter* must be a parameter of an *action* that must be in the *plan*, and that the describing *proposition* involves *parameter*. The description should be sufficient to allow the hearer to execute the action, all other things being equal. This effect on the hearer is summarized by the assertion

KNOW-PARAMETER (hearer, parameter, action, plan).

For example, if agent P knows how many dollars a certain ticket sells for, then

KNOW-PARAMETER (P, AMT,
GIVE-MONEY (P, AMT), BUY-TICKET (P, C, TKT))

is true. In other words, the agent P knows a description of AMT sufficient to be able to perform the GIVE-MONEY act in the BUY-TICKET plan. Though the axiomatization of KNOW-PARAMETER is problematic, we will be using it only in simple cases where its use is straightforward.

INTRODUCE-PLAN introduces a plan into a conversation. In particular, it allows a discourse intention to just refer to a plan (rather than refer to a plan by executing it, clarifying it, and so on). INTRODUCE-PLAN is performed by stating a particular goal of the speaker (the decomposition), where the goal is a subgoal of *plan* (the constraint). The effect is that the hearer is then aware of the speaker's plan.

3.3 *Discourse Intentions*

Discourse intentions are purposes of the speaker, expressed in terms of both the task plans of the speaker (the domain plans) and the plans recursively generated by these plans (the discourse plans). We will limit ourselves to two types of discourse intentions: those in which agents intend to do these plans, and those in which agents intend other agents to intend to do these plans. We will use the speech act REQUEST to represent the latter case.[7]

Though the recognition of a *structure* of discourse intentions (Grosz and Sidner 1986) plays an important role in the recognition of discourse structure, this problem is not considered here. As discussed above, we are concerned with the recognition of structures of commonsense plans, to which individual discourse intentions refer.

3.4 *Speech Act Definitions*

To model speech acts (Searle 1969), we are assuming plan-based definitions as in Allen and Perrault 1980. In particular, figure 17.4 shows the formulation of the speech acts REQUEST, INFORM, and INFORMREF (a variation needed to handle "wh"-questions). INFORMIF, needed to handle yes-no questions, will not be needed for later examples and is thus not shown. We are also omitting our treatment (Litman and Allen 1987) of indirect speech acts (Searle 1975).

Unlike Allen and Perrault, we have explicitly added an effect that the hearer then believes the preconditions held if the act is done successfully. This could be inferred from first principles, but adding it to the definition allows us to use a simple plan recognition algorithm throughout the paper.

The only other difference from Allen and Perrault 1980 is that we have added an extra parameter to the INFORMREF action and the KNOWREF assertion. The assertion KNOWREF (A, t, p) means that A knows a description of term t, which satisfies proposition p and is informative enough to execute a plan.

This is simply a notational variant that is closer to the actual implementation. Thus, rather than stating the goal to know when train TR1 leaves as

KNOWREF (A, the x: depart-time (TR1, x)),

as in Allen and Perrault 1980, we write

KNOWREF (A, ?time, EQUAL (depart-time (TR1), ?time)).

Not all such assertions involve the equality predicate. For example, the representation of the goal behind the utterance "What do you want me to do?" would be

KNOWREF (speaker, ?speaker-action, WANT (hearer, ?speaker-action)).

We can define this operator formally within a possible-worlds semantics of the BELIEF operator by using "quantifying in" as done in Allen and Perrault 1980. Although this analysis is not fully satisfactory, it is adequate for our present purposes.

HEADER:	REQUEST(speaker, hearer, action)
PREREQUISITE:	WANT(speaker, action)
DECOMPOSITION:	SURFACE-REQUEST(speaker, hearer, action)
EFFECTS:	WANT(hearer, action) KNOW(hearer, WANT(speaker, action))
CONSTRAINT:	AGENT(action, hearer)
HEADER:	INFORM(speaker, hearer, proposition)
PREREQUISITE:	KNOW(speaker, proposition)
DECOMPOSITION:	SURFACE-INFORM(speaker, hearer, proposition)
EFFECTS:	KNOW(hearer, proposition) KNOW(hearer, KNOW(speaker, proposition))
HEADER:	INFORMREF(speaker, hearer, term, proposition)
PREREQUISITE:	KNOWREF(speaker, term, proposition)
DECOMPOSITION:	achieve KNOW(hearer, proposition)
EFFECT:	KNOWREF(hearer, term, proposition)
CONSTRAINT:	PARAMETER(term, proposition)

Figure 17.4
Speech act definitions.

3.5 *An Example*

To illustrate the relationships between discourse intentions, domain plans, and discourse plans generated from domain plans, consider again the first few utterances of Dialogue 6, repeated below for convenience:

(1) Person: Where do I get the CAN-AM train?

(2) Clerk: It's not until six.
(3) They're meeting at cubicle C.
(4) Person: Where's that?

Using the plan schemas defined above, we can represent the discourse intentions behind these utterances informally as follows:

(1) Person intends that Clerk intends that
Clerk identify the location parameter
of the GOTO step of a BOARD CAN-AM train plan.

(2) Clerk intends that
Clerk identify the time parameter
of the GOTO step of a BOARD CAN-AM train plan.

(3) Clerk intends that
Clerk identify the location parameter
of the GOTO step of a BOARD CAN-AM train plan.

(4) Person intends that Clerk intends that
Clerk identify a parameter
of the identify-parameter of (3) above.

Note that each discourse intention refers to a discourse plan,[8] which itself refers to its generating domain (as in (1), (2), (3)) or discourse (as in (4)) plan (and so on recursively, for discourse plans). In other words, each discourse intention ultimately points to a portion of a commonsense domain plan instantiation structure.

More formally, let us reconsider the intention behind utterance (1). We will show later that this intention refers to the discourse plan

IDENTIFY-PARAMETER(Clerk, Person, location(CAN-AM train),
GOTO(Person, location(CAN-AM train), time(CAN-AM train)),
TAKE-TRAIN-TRIP(Person, CAN-AM train, destination (CAN-AM train))),

which itself refers to the GOTO subplan,

GOTO(Person, location(CAN-AM train), time(CAN-AM train)),

of the (BOARD subplan of the) domain plan

TAKE-TRAIN-TRIP(Person, CAN-AM train, destination (CAN-AM train)).

Figure 17.5 sketches this situation. In the next section we discuss how to recognize such commonsense plan structures.

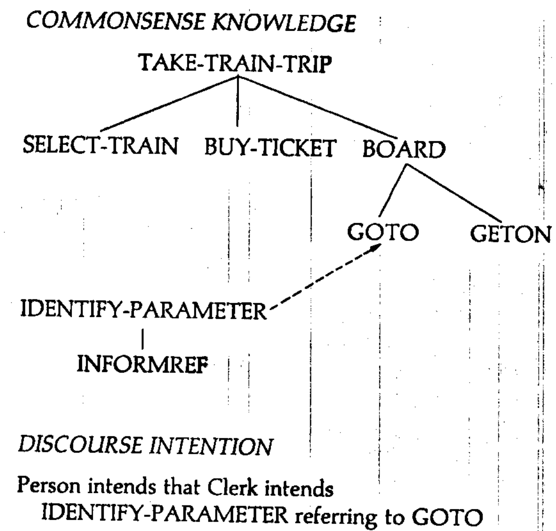

Figure 17.5
Example analysis: Where do I get the CAN-AM train?

4 *Domain and Discourse Plan Recognition*

4.1 *Overview*

The task of the plan recognizer is to recognize a domain plan structure, and any discourse plans generated by this structure, to which the discourse intentions of an input utterance refer. That is, the plan recognizer's task is not only to infer a domain plan (as in previous systems) but also to infer any plans generated by this domain plan that are relevant to the discourse intentions. The plan recognizer has at its disposal a library of domain plan schemas varying with the domain and a library of discourse and speech act plan schemas (as in figures 17.2–17.4), as well as the representation of the parse of the input utterance. The plan recognizer will use this information to associate each utterance with an intention referring to a domain plan, either directly or through a series of generated discourse plans. If the interpretation is ambiguous, a discourse intention and set of commonsense plans are created for each interpretation.

4.2 *Plan Chaining and Constraint Satisfaction*

The plan recognizer builds a domain plan by finding a sequence of instantiations of domain plan schemas, each one containing the previous one in its decomposition,[9] beginning with a plan schema connected to the surface speech analysis produced by the parser. In the simplest case, the speech act matches the decomposition of a domain plan or is a request for an act that matches the decomposition of a domain plan. In more complicated cases, the speech act is connected to a domain plan by a sequence of discourse plans related to the domain plan. Here, the speech act matches the decomposition of a discourse plan or is a request for an act that matches the decomposition of a discourse plan.

Since every discourse plan takes another plan as an argument, when any discourse plan is recognized from a speech act, an associated domain plan will implicitly be introduced. Furthermore, the constraints relating the discourse plan to this domain plan add information about the domain plan enabling further plan recognition. For instance, utterance (1) in Dialogue 4—"I'd like to buy a ticket to New York"—could be identified as a part of discourse plan to introduce a plan, with constraints that specify that buying a ticket is a step of the plan being introduced. To constrain the plan being introduced in this way, we can invoke the plan recognizer recursively, starting with buying a ticket as the observed action. For example, this act could be a step in a plan to take a train trip. Since taking a trip is a domain plan, no other plans are introduced and recursive plan recognition halts.

Once a set of discourse and domain plans is recognized, each is expanded top down by adding the definitions of all steps and substeps (based on the plan libraries), until there are no unique expansions for any of the remaining substeps. For example, once TAKE-TRAIN-TRIP is recognized from BUY-TICKET, it is expanded to include SELECT-TRAIN and BOARD as well (recall figure 17.2).

Though plan chaining is a simple tree search and thus theorectially terminates, if unconstrained it can be both costly and unable to yield a unique plan interpretation. The search process needs to be controlled with various heuristics and limited by dividing it into incremental stages.

4.3 *Coherence Heuristics*

The search process described above does not take into account the influence of the active discourse context.[10] One coherence heuristic controls the plan recognition process by preferring the recognition of plans that are related to currently active plans, rather than to plans constructed from first principles. That is, portions of the plans referred to by a current discourse intention may have already been recognized as objects of previous discourse intentions.

Furthermore, when several discourse and domain plans can be related to currently active plans, other coherence heuristics place a further preference ordering on the recognition process. (When there are no active plans, as at the beginning of a dialogue, the plan recognizer applies the heuristics to

the library of domain plan schemas instead.) In particular:

1. The recognition of steps in domain plans or in continuation discourse plans is attempted first. If the utterance can be viewed as intending (or requesting that another intend) a linguistic step in an already intended plan, or in a continuation discourse plan, then it is classified as such without further consideration. The speaker is believed to be doing exactly what is expected in the given situation.
2. Clarification plans are attempted next. If the utterance cannot be viewed as a continuation but can be viewed as a clarification discourse plan generated by an intended plan, then it is classified as such without further consideration.

Thus, the plan recognizer prefers an utterance interpretation that directly continues rather than clarifies (that is, indirectly refers to) a plan.

Although these heuristics and their ordering have not been validated with psychological experimentation, they have intuitive appeal. For example, preferring current plans over plans constructed from first principles, and continuations of these plans over clarifications, allows the preferred discourse behavior to default to the earlier plan-based systems in which a dialogue contains no topic changes or unexpected subdialogues. The heuristics also follow the principle of Occam's razor, since they are ordered so as to introduce as few new plans as possible.

4.4 *Linguistic Clues*

In many models of discourse (Grosz 1977; Reichman-Adar 1984; Cohen 1983; Grosz and Sidner 1986) rules of conversational coherence have been shown to govern the use of surface linguistic phenomena such as cue phrases and choice of reference. Thus, in addition to coherence heuristics, the plan recognizer can also use surface linguistic phenomena to control its search. In particular, the information conveyed by such phenomena takes precedence over the coherence heuristics used by the plan recognizer in the linguistically unmarked case.

For example, consider recognizing a clarification that is not prefaced by a cue phrase. Using the coherence preferences, the plan recognizer first tries to intepret the utterance as a continuation, and only when these efforts fail, as a clarification. This is because a clarification is less expected in the unmarked case. However, if a clarification class discourse plan is prefaced with a cue phrase such as "no," a continuation is not tried first. This is because a signal for a plan such as a correction is explicitly present. (As in other models of discourse, we are assuming that cue uses of lexical items can be distinguished from other uses, for example by intonation (Hirschberg and Litman 1987).) Cues thus ease the recognition of relationships that would otherwise be difficult to understand.

4.5 *Plan-Based Heuristics*

Although the plan recognizer may suggest a plan as a linguistically coherent interpretation, the plan may still be eliminated from further consideration by a set of heuristics based on rational planning behavior as in Allen and Perrault 1980. For example, plans that are postulated whose effects are already true are eliminated, as are plans whose constraints cannot be satisfied.

4.6 *Incremental Search*

When the heuristics and linguistic constraints cannot eliminate all but one postulated plan, the chaining stops, even when a solution path has not yet been found. In other words, search terminates after branch points. For example, since BOARD and MEET plans can be recognized from a GOTO (recall figure 17.2), chaining would halt with two branches if both plans were plausible. Such premature termination is typical in dialogue processing (see, for example, Carberry 1985), since later utterances in a dialogue often eliminate many of the branches. Cohen, Perrault, and Allen (1982) also argue that rational speakers would not have intended inferences in branches since such inferences could not have been drawn unambiguously. Thus, the plan recognition process halts either when we have uniquely incorporated the utterance into our expectation structure as defined by our heuristics or when the search space becomes too large. In the latter case the plan remains ambiguous.

5 *An Example*

This section uses the framework developed in the previous sections to illustrate the recognition of the commonsense domain and discourse plans underlying a dialogue.[11] Although the behavior to be described is fully specified by the theory, our implementation corresponds only to the model of plan recognition from first principles. Simulated computational processes have been implemented elsewhere, however. For example, the current system assumes that a highly specialized semantic grammar (Brown and Burton 1977) has parsed the utterances in the train domain. This allows the avoidance of some difficult parsing issues and concentration on the plan recognition model. Though such a grammar has been implemented, it is currently not hooked up to the plan recognition system. Litman 1985 contains a full discussion of the implementation and associated issues of knowledge representation. Also, the theory at present does not concern itself with the planning or generation of natural-language responses. The examples will describe what the system should do (using the actual response in the dialogue as a guide).

As our example, let us return again to the initial utterances of Dialogue 6:

Person:	Where do I get the CAN-AM train?
Clerk:	It's not until six.
	They're meeting at cubicle C.
Person:	Where's that?

The initial state of the plan recognition system consists of a library of speech acts, discourse plans, and domain plans regarding trains (that is, plans such as those found in figures 17.2–17.4). Upon hearing the initial utterance, the following simulated parse is also input to the system:

SURFACE-REQUEST(Person1, Clerk1,
  I1:INFORMREF(Clerk1, Person1, ?term, EQUAL
   (?term, loc(dtrain1)))).

A SURFACE-REQUEST for an INFORMREF is recognized from the interrogative mood of the utterance, as in Allen and Perrault 1980. The type of *dtrain1*, the train referred to by "the CAN-AM train," is restricted to a departing train using the verb "get." The INFORMREF action will be referred to using the name "I1."

Since we are at the beginning of a dialogue, the plan recognizer uses its general plan schemas rather than previously recognized plan instantiations to construct its analysis. According to coherence preference (1), the plan recognizer first checks to see whether the SURFACE-REQUEST matches (or is a request for an action that matches) the decomposition of a domain plan schema, or matches a continuation discourse plan referring to a domain plan. These attempts fail, however.

According to coherence preference (2), the plan recognizer then checks to see whether the SURFACE-REQUEST can be related instead to a clarification discourse plan schema. This time there is a successful match: Person is requesting Clerk to perform an action, I1, that matches the decomposition of the IDENTIFY-PARAMETER clarification discourse plan.

Before pursuing the candidate IDENTIFY-PARAMETER plan any further, the plan recognizer checks on the plan's reasonableness using the plan-based heuristics. In satisfying the constraints on IDENTIFY-PARAMETER (Clerk1, Person1, ?parameter, ?action, ?plan), that is,

1. PARAMETER(?parameter, ?action)
2. STEP(?action, ?plan)
3. PARAMETER(?parameter, EQUAL(?parameter, loc(dtrain1))),

a second plan is introduced (?plan) that must have a step ?action that contains the location of a train, described via the equality of the INFORMREF. The only eligible plan for ?action is GOTO. A new plan is

COMMONSENSE KNOWLEDGE

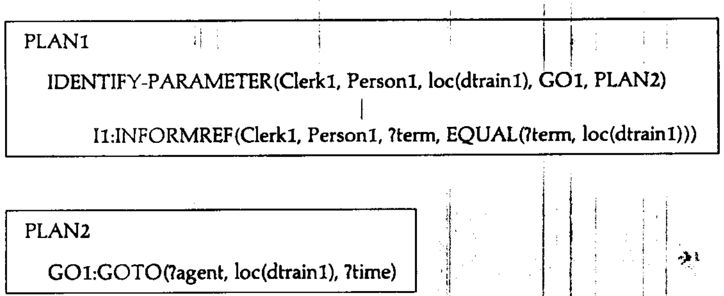

Figure 17.6
Chaining and constraint satisfaction produce intermediate plan structures.

thus created containing GOTO and arbitrarily called PLAN2. This state of affairs is shown in figure 17.6, where the name of a plan structure appears at the top left-hand corner. The plan recognizer also verifies that the effects of the recognized plans are not already satisfied.

The plan recognition algorithm is now recursively called on this GOTO. Decomposition chaining yields both the BOARD and MEET plans. The MEET plan is eliminated due to typing constraint violation; dtrain1 is already known to be a departing train from the parse. Since the expected agent of the BOARD plan is the speaker, ?agent is heuristically set equal to Person1. Finally, chaining proceeds from BOARD to TAKE-TRAIN-TRIP, and since TAKE-TRAIN-TRIP is a domain plan (that is, no more plans are introduced), plan recognition ends. The plans are expanded top down to include the rest of their steps, with the final structures shown in figure 17.7. (The dotted lines indicate the information inferred from the top-down chaining.)

Note that the plan recognizer has constructed a domain plan structure and a discourse plan generated by this structure, to which the discourse intention of the utterance (Person's REQUEST) refers. Loosely, this discourse intention is that Person intends that Clerk intends to clarify a part of Person's domain plan. (Although we have now abstracted discourse structure (as opposed to commonsense plan) processing out of our model, this discourse intention could then be manipulated by attentional and intentional processing, as in previous formulations of our work (Litman and Allen 1987) or as in Grosz and Sidner 1986.) Note too that the domain plan structure contains more domain information than explicitly referred to by the discourse intention. The necessity for this "extra" domain plan recognition was discussed earlier in the paper.

COMMONSENSE KNOWLEDGE

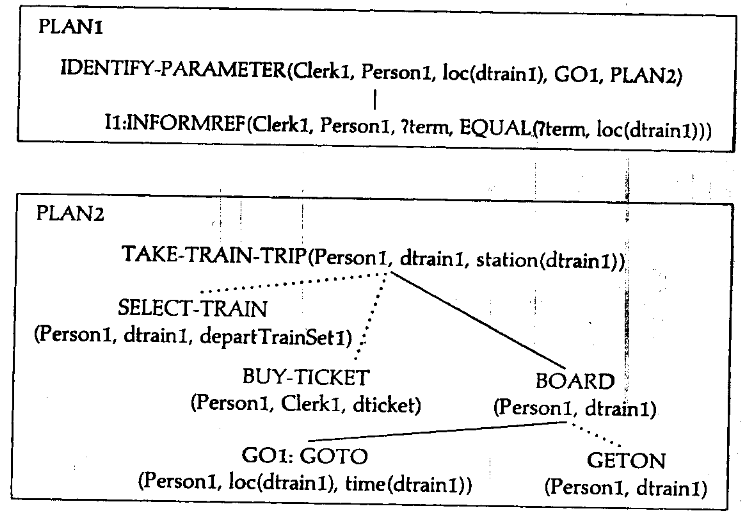

Figure 17.7
Plans recognized after the first utterance.

Clerk's planning of the response is not specified in this theory. What the system should do will be sketched here. The system, examining the commonsense plans and discourse information, responds with both a helpful response ("It's not until six") and the expected clarification ("They're meeting at cubicle C"). These utterances execute a new clarification discourse plan IDENTIFY-PARAMETER (Clerk1, Person1, time(dtrain1), GO1, PLAN2), which we will call PLAN3, and the existing discourse plan, PLAN1. Thus, at the stage just before Person1 speaks again, the commonsense knowledge base would contain PLAN1, PLAN2, PLAN3.

Person then asks "Where's that?", which would be parsed into the linguistic action as follows:

SURFACE-REQUEST(Person1, Clerk1
    INFORMREF(Clerk1, Person1, ?term1,
    EQUAL(?term1, loc(CubicleC)))).

This analysis assumes the appropriate resolution of "that" to CubicleC by an immediate focus mechanism such as Sidner's (1983). This makes the example simpler. In this case the plan recognition would work even if the "that" were not resolved and left as an unknown constant, as discussed below.

Although Clerk thought the utterance executing PLAN1 achieved the desired Person KNOW-PARAMETER, in actuality it did not provide a description enabling Person to execute the GOTO of PLAN2. Instead we have another request for clarification. The plan recognizer attempts to relate this utterance to the active commonsense plans based on the coherence heuristics. The first preference fails since the SURFACE-REQUEST does not execute or continue any of the steps in TAKE-TRAIN-TRIP (which should have been resumed if the previous clarifications were understood), or in plans 1 and 3 (since they are completed). The second preference succeeds, and the utterance is recognized as requesting a new clarification, IDENTIFY-PARAMETER (Clerk1, Person1, loc(Cubicle C), S-I, PLAN1), where S-I is the SURFACE-INFORM representing the clerk's realization "They're meeting at cubicle C," of I1. This process is basically analogous to the process discussed in detail above, except that PLAN1 to which the IDENTIFY-PARAMETER refers is found rather than constructed. Note, too, that the plan to which the IDENTIFY-PARAMETER refers is now a discourse plan rather than a domain plan.

As mentioned above, if in the input "that" was originally unresolved (that is, CubicleC is unknown), "that" will be correctly resolved later as a side effect of the plan recognition. In particular, the appropriate binding will be made during the constraint propagation process connecting the current IDENTIFY-PARAMETER to PLAN1.

6 Summary

We have presented a model for recognizing domain task plans underlying task-oriented dialogues, as well as discourse tasks generated by these domain tasks, to which a discourse can also refer. Our model accounts for subdialogues generated by a task, such as clarifications and corrections, yet maintains the computational advantages of the plan-based approach.

Commonsense knowledge about generated discourse interactions was incorporated into a planning framework using the same plan-based formalism previously used for representing domain knowledge. Such knowledge was needed to mediate between task plans and many discourse intentions. This new knowledge was formalized using a set of discourse plans, plans explicitly encoding indirect relationships between discourse intentions and domain plans. This formulation allowed us to easily recognize discourse plans of domain plans, as well as discourse plans generated by other discourse plans to an arbitrary level of nesting.

Domain and discourse plans were recognized and related by a context-dependent, heuristic plan recognition algorithm. First, a discourse or domain plan was recognized from every utterance. The recognition of this plan was constrained by preferring continuations of expected plans over clarifications, and already existing plans over plans constructed from first principles. If the recognized plan was a discourse plan, the formal constraints specifying the discourse and domain plan relationships were then used to recursively generate recognition of the domain plan from recognition of the discourse plan. These constraints provided the link between what one indirectly said in a discourse about a task and the commonsense knowledge about the task itself.

Notes

1. Task-oriented dialogues occur when the topic of a dialogue is related to a particular domain task (for instance, take a train trip). As such, our task-oriented dialogues are a generalization of the "task-oriented" dialogues of Grosz (1977) (dialogues in which two people work cooperatively on a task that is performed during the dialogue) and the "information-seeking" dialogues of Carberry (1985) (dialogues in which an agent seeks information with respect to a task that is not executed during the dialogue).
2. Commonsense knowledge refers to the intuitive beliefs and theories about the world used by many artificial intelligence systems.
3. Although we have taken a "data-structure view of plans" (Pollack 1986), even if one takes a "mental phenomenon view of plans" (Pollack 1986), discourse intentions are still not identical to plans (Grosz and Sidner 1986).
4. Because "discourse" plans may in theory account for nonlinguistic actions, as well as for reasons discussed below, we originally called such plans "metaplans" (Litman 1985). We changed terminology, however, since "metaplan" has various other uses in the literature (see, for example, Wilensky 1983), and to date we have only been concerned with linguistically realized metaplans.
5. Wilensky (1983) has also discussed the role of metaplans in plan-based natural-language understanding systems. However, there are several important differences between Wilensky's work and the work reported here. First, Wilensky's metaplans are concerned with interactions between multiple plans and are used to handle issues of concurrent goal interaction. The metaplans of this work are concerned with manipulations of single plans and are used to handle issues of discourse. Moreover, Wilensky does not specify representation or recognition details (for example, the mechanisms relating metaplan and nonmetaplan recognition), issues that we explicitly address in this paper.
6. This formulation of IDENTIFY-PARAMETER is a simplification of the version found in Litman and Allen 1987. It is sufficient for the examples covered in this paper.
7. In previous formulations of our work (Litman and Allen 1987) INTRODUCE-PLAN was also used to represent this latter type of discourse intention. By making explicit the separation between discourse intentions and commonsense plans, the two situations can now be clearly distinguished.
8. It is important to remember that although in this dialogue all discourse intentions refer to the domain plans indirectly, direct reference is also possible. An example would be the intention behind the utterance "Person: Sell me a ticket to New York"—that is, Person intends that Clerk intends that Person BUY-TICKET to New York.
9. Plan chaining can also be done via effects and preconditions. Pollack (1986) even allows recognition of nonexistent library plans. To keep the examples simple, all plan schemas have been expressed so that chaining via decomposition is sufficient.
10. In this paper we assume the existence of a separate attentional component of discourse structure (Grosz and Sidner 1986). Such a component abstracts the focus of attention of

the discourse participants. See Litman and Allen 1987 for a previous formulation of our work that directly incorporated such a component in the form of a plan stack.

11. For the purposes of this paper, we limit ourselves to discourse intentions recognized from single utterances. Litman and Allen 1987 illustrates the treatment of a restricted class of multiple utterances.

References

Allen, J. F., and C. R. Perrault (1980). Analyzing intention in utterances. *Artificial Intelligence* 15, 143–178.

Brown, J., and R. Burton (1977). Semantic grammar: A technique for constructing natural language interfaces to instructional systems. Technical Report 3587, Bolt Beranek and Newman, Inc., Cambridge, MA.

Carberry, M. S. (1985). Pragmatic modeling in information system interfaces. Doctoral dissertation and Technical Report 86-07 (1986), Department of Computer and Information Sciences, University of Delaware, Newark, DE.

Cohen, P. R., C. R. Perrault, and J. F. Allen (1982). Beyond question answering. In W. Lehnert and M. Ringle, eds., *Strategies for natural language processing*. Hillsdale, NJ: L. Erlbaum Associates.

Cohen, R. (1983). A computational model for the analysis of arguments. Technical Report CSRG-151 and Doctoral dissertation, University of Toronto, Toronto, Ont.

Fikes, R. E., and N. J. Nilsson (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence* 2, 189–208.

Grosz, B. J. (1977). The representation and use of focus in dialogue understanding. Technical Note 151, Artificial Intelligence Center, SRI International, Menlo Park, CA.

Grosz, B. J., and C. L. Sidner (1986). Attention, intentions, and the structure of discourse. *Computational Linguistics* 12, 175–204.

Hirschberg, J., and D. Litman (1987). Now let's talk about 'now': Identifying cue phrases intonationally. In *Proceedings of the Twenty-fifth Annual Meeting*, Association for Computational Linguistics, Stanford, CA.

Horrigan, M. K. (1977). Modelling simple dialogs. Master's thesis and Technical Report 108, University of Toronto, Toronto, Ont.

Litman, D. J. (1985). Plan recognition and discourse analysis: An integrated approach for understanding dialogues. Doctoral dissertation and Technical Report 170, University of Rochester, Rochester, NY.

Litman, D. J., and J. F. Allen (1987). A plan recognition model for subdialogues in conversation. *Cognitive Science* 11, 163–200.

Pollack, M. E. (1986). A model of plan inference that distinguishes between the beliefs of actors and observers. In *Proceedings of the Twenty-fourth Annual Meeting*, Association for Computational Linguistics, New York, NY.

Reichman-Adar, R. (1984). Extended person-machine interfaces. *Artificial Intelligence* 22, 157–218.

Sacerdoti, E. D. (1977). *A structure for plans and behavior*. New York: American Elsevier.

Searle, J. R. (1969). *Speech acts: An essay in the philosophy of language*. New York: Cambridge University Press.

Searle, J. R. (1975). Indirect speech acts. In P. Cole and J. Morgan, eds., *Syntax and semantics 3: Speech acts*. New York: Academic Press.

Sidner, C. L. (1983). Focusing in the comprehension of definite anaphora. In M. Brady and R. C. Berwick, eds., *Computational models of discourse*. Cambridge, MA: MIT Press.

Sidner, C. L. (1985). Plan parsing for intended response recognition in discourse. *Computational Intelligence* 1, 1–10.

Wilensky, R. (1983). *Planning and understanding*. Reading, MA: Addison-Wesley.

Chapter 18

Communicative Intentions, Plan Recognition, and Pragmatics: Comments on Thomason and on Litman and Allen

Kent Bach

Like others concerned with discourse and plan recognition, both Thomason and Litman and Allen appreciate the importance of the audience's recognition of the speaker's intentions and plans. Yet they ovelook certain important concepts and distinctions from recent work in the theory of speech acts. I am not suggesting that this oversight leaves their approaches with insoluble problems; rather, I am suggesting that their approaches can benefit by taking these notions into account. After all, plans are systems of intentions, and speech act theory is concerned with speaker's intentions and their recognition. However, there are many faces to a speaker's intentions, and the role and even the relevance of the audience's recognition of an intention depends on the kind of intention it is. In order to show this, I will sketch a theoretical framework within which to comment on the target papers.

1 *Theoretical Framework*

In general, a person's behavior counts as an action (or at least an attempted action) of a certain sort in virtue of his intention. Generally it makes no difference whether any witnesses happen to identify this intention, much less whether the agent intends them to do so. For example, your bodily movements count as practicing break dancing no matter what an observer thinks, or what you intend him to think, you are doing. However, in the case of an utterance, where there is a presumption of audience-directed communication, the audience has reason to think that the speaker intends him to recognize this intention, that is, to identify what he is doing in making the utterance. Moreover, the intention for him to recognize this intention is not a distinct intention; rather, it is the very intention that he is to recognize. This is the intention that constitutes the utterance as an act of communication, as an *illocutionary* act with a certain force and content. As such, it succeeds just in case the hearer recognizes that intention.

However, in saying something, a speaker is likely to have intentions of several different sorts. One hardly ever intends merely to communicate: