

# A Plan Recognition Model for Clarification Subdialogues

Diane J. Litman and James F. Allen  
Department of Computer Science  
University of Rochester, Rochester, NY 14627

## Abstract

One of the promising approaches to analyzing task-oriented dialogues has involved modeling the plans of the speakers in the task domain. In general, these models work well as long as the topic follows the task structure closely, but they have difficulty in accounting for clarification subdialogues and topic change. We have developed a model based on a hierarchy of plans and metaplans that accounts for the clarification subdialogues while maintaining the advantages of the plan-based approach.

## 1. Introduction

One of the promising approaches to analyzing task-oriented dialogues has involved modeling the plans of the speakers in the task domain. The earliest work in this area involved tracking the topic of a dialogue by tracking the progress of the plan in the task domain [Grosz, 1977], as well as explicitly incorporating speech acts into a planning framework [Cohen and Perrault, 1979; Allen and Perrault, 1980]. A good example of the current status of these approaches can be found in [Carberry, 1983]. In general, these models work well as long as the topic follows the task structure closely, but they have difficulty in accounting for clarification subdialogues and topic change.

Sidner and Israel [1981] suggest a solution to a class of clarification subdialogues that correspond to debugging the plan in the task domain. They allow utterances to talk about the task plan, rather than always being a step in the plan. Using their suggestions, as well as our early work [Allen et al., 1982; Litman, 1983], we have developed a model based on a hierarchy of plans and metaplans that

-----  
This work was supported in part by the National Science Foundation under Grant IST-8210564, the Office of Naval Research under Grant N00014-80-C-1097, and the Defense Advanced Research Projects Agency under Grant N00014-82-K-0193.

accounts for the debugging subdialogues they discussed, as well as other forms of clarification and topic shift.

Reichman [1981] has a structural model of discourse that addresses clarification subdialogues and topic switch in unconstrained spontaneous discourse. Unfortunately, there is a large gap between her abstract model and the actual processing of utterances. Although not the focus of this paper, we claim that our new plan recognition model provides the link from the processing of actual input to its abstract discourse structure. Even more important, this allows us to use the linguistic results from such work to guide and be guided by our plan recognition.

For example, consider the following two dialogue fragments. The first was collected at an information booth in a train station in Toronto [Horrigan, 1977], while the second is a scenario developed from protocols in a graphics command and control system that displays network structures [Sidner and Bates, 1983].

- 1) Passenger: The eight-fifty to Montreal?
- 2) Clerk: Eight-fifty to Montreal. Gate seven.
- 3) Passenger: Where is it?
- 4) Clerk: Down this way to the left. Second one on the left.
- 5) Passenger: OK. Thank you.

### Dialogue 1

- 6) User: Show me the generic concept called "employee."
- 7) System: OK. <system displays network>
- 8) User: I can't fit a new IC below it. Can you move it up?
- 9) System: Yes. <system displays network>
- 10) User: OK, now make an individual employee concept whose first name is "Sam" and whose last name is "Jones." The Social Security number is 234-56-7899.
- 11) System: OK.

### Dialogue 2

While still "task-oriented," these dialogues illustrate phenomena characteristic of spontaneous conversation. That is, subdialogues correspond not only to subtasks (utterances (6)-(7) and (10)-(11)), but also to clarifications ((3)-(4)), debugging of task execution ((8)-(9)), and other types of topic switch and resumption. Furthermore, since these are extended discourses rather than unrelated question/answer exchanges, participants need to use the information provided by previous utterances. For example, (3) would be difficult to understand without the discourse context of (1) and (2). Finally, these dialogues illustrate the following of conversational conventions such as terminating dialogues (utterance (5)) and answering questions appropriately. For example, in response to (1), the clerk could have conveyed much the same information with "The departure location of train 537 is gate seven," which would not have been as appropriate.

To address these issues, we are developing a plan-based natural language system that incorporates knowledge of both task and discourse structure. In particular, we develop a new model of plan recognition that accounts for the recursive nature of plan suspensions and resumptions. Section 2 presents this model, followed in Section 3 by a brief description of the discourse analysis performed and the task and discourse interactions. Section 4 then traces the processing of Dialogue 1 in detail, and then this work is compared to previous work in Section 5.

## 2. Task Analysis

### 2.1 The Plan Structures

In addition to the standard domain-dependent knowledge of task plans, we introduce some knowledge about the planning process itself. These are domain-independent plans that refer to the state of other plans. During a dialogue, we shall build a stack of such plans, each plan on the stack referring to the plan below it, with the domain-dependent task plan at the bottom. As an example, a clarification subdialogue is modeled by a plan structure that refers to the plan that is the topic of the clarification. As we shall see, the manipulations of this stack of plans is similar to the manipulation of topic hierarchies that arise in discourse models.

To allow plans about plans, i.e., *metaplans*, we need a vocabulary for referring to and describing plans.

Developing a fully adequate formal model would be a large research effort in its own right. Our development so far is meant to be suggestive of what is needed, and is specific enough for our preliminary implementation. We are also, for the purpose of this paper, ignoring all temporal qualifications (e.g., the constraints need to be temporally qualified), and all issues involving beliefs of agents. All plans constructed in this paper should be considered mutually known by the speaker and hearer.

We consider plans to be networks of actions and states connected by links indicating 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. Associated with each plan is a set of *constraints*, which are assertions about the plan and its terms and parameters. The use of constraints will be made clear with examples. As usual, plans may also contain *prerequisites*, *effects*, and a *decomposition*. Decompositions may be sequences of actions, sequences of subgoals to be achieved, or a mixture of both. We will ignore most prerequisites and effects throughout this paper, except when needed in examples.

For example, the first plan in Figure 1 summarizes a simple plan schema with a header "BOARD (agent, train)," with parameters "agent" and "train," and with the constraint "depart-station (train) = Toronto." This constraint captures the knowledge that the information booth is in the Toronto station. The plan consists of the three steps indicated. The prerequisites and effects are not

```

HEADER: BOARD (agent, train)
STEPS: do BUY-TICKET (agent, train)
        do GOTO (agent, depart-location (train),
                depart-time (train))
        do GETON (agent,train)
CONSTRAINTS: depart-station (train) = Toronto
-----
HEADER: GOTO (agent, location, time)
EFFECT: AT (agent, location, time)
-----
HEADER: MEET (agent, train)
STEPS: do GOTO (agent, arrive-location (train),
                arrive-time (train))
CONSTRAINTS: arrive-station (train) = Toronto
-----

```

Figure 1: Domain Plans

shown. The second plan indicates a primitive action and its effect. Other plans needed in this domain would include plans to meet trains, plans to buy tickets, etc.

We must also discuss the way terms are described, for some descriptions of a term are not informative enough to allow a plan to be executed. What counts as an informative description varies from plan to plan. We define the predicate KNOWREF (agent, term, plan) to mean that the agent has a description of the specified term that is informative enough to execute the specified plan, all other things being equal. Throughout this paper we assume a typed logic that will be implicit from the naming of variables. Thus, in the above formula, *agent* is restricted to entities capable of agency, *term* is a description of some object, and *plan* is restricted to objects that are plans.

Plans about plans, or metaplans, deal with specifying parts of plans, debugging plans, abandoning plans, etc. To talk about the structure of plans we will assume the predicate *IS-PARAMETER-OF* (*parameter*, *plan*), which asserts that the specified parameter is a parameter of the specified plan. More formally, parameters are skolem functions dependent on the plan.

Other than the fact that they refer to other plans, metaplans are identical in structure to domain plans. Two examples of metaplans are given in Figure 2. The first one, SEEK-ID-PARAMETER, is a plan schema to find out a suitable description of the parameter that would allow the plan to be executed. It has one step in this version, namely to achieve KNOWREF (agent, parameter, plan), and it has two constraints that capture the relationship between the metaplan and the plan it concerns, namely that "parameter" must be a parameter of the specified plan, and that its value must be presently unknown.

The second metaplan, ASK, involves achieving KNOWREF (agent, term, plan) by asking a question and receiving back an answer. Another way to achieve KNOWREF goals would be to look up the answer in a reference source. At the train station, for example, one can find departure times and locations from a schedule.

We are assuming suitable definitions of the speech acts, as in Allen and Perrault [1980]. The only deviation from that treatment involves adding an extra argument onto each (nonsurface) speech act, namely a plan parameter that provides the context for the speech act. For

```

-----
HEADER: SEEK-ID-PARAMETER (agent, parameter,
                           plan)
STEPS: achieve KNOWREF (agent, parameter, plan)
CONSTRAINTS: IS-PARAMETER-OF (parameter, plan)
              ~KNOWREF (agent, parameter, plan)
-----
HEADER: ASK (agent, term, plan)
STEPS: do REQUEST (agent, agent2,
                  INFORMREF (agent2, agent, term, plan),
                  plan)
        do INFORMREF (agent2, agent, term, plan)
EFFECTS: KNOWREF (agent, term, plan)
CONSTRAINTS: ~KNOWREF (agent, term, plan)
-----

```

**Figure 2: Metaplans**

example, the action INFORMREF (agent, hearer, term, plan) consists of the agent informing the hearer of a description of the term with the effect that KNOWREF (hearer, term, plan). Similarly, the action REQUEST (agent, hearer, act, plan) consists of the agent requesting the hearer to do the act as a step in the specified plan. This argument allows us to express constraints on the plans suitable for various speech acts.

There are obviously many more metaplans concerning plan debugging, plan specification, etc. Also, as discussed later, many conventional indirect speech acts can be accounted for using a metaplan for each form.

## 2.2 Plan Recognition

The plan recognizer attempts to recognize the plan(s) that led to the production of the input utterance. Typically, an utterance either extends an existing plan on the stack or introduces a metaplan to a plan on the stack. If either of these is not possible for some reason, the recognizer attempts to construct a plausible plan using any plan schemas it knows about. At the beginning of a dialogue, a disjunction of the general expectations from the task domain is used to guide the plan recognizer.

More specifically, the plan recognizer attempts to incorporate the observed action into a plan according to the following preferences:

- 1) by a direct match with a step in an existing plan on the stack;

- 2) by introducing a plausible subplan for a plan on the stack;
- 3) by introducing a metaplan to a plan on the stack;
- 4) by constructing a plan, or stack of plans, that is plausible given the domain-specific expectations about plausible goals of the speaker.

Class (1) above involves situations where the speaker says exactly what was expected given the situation. The most common example of this occurs in answering a question, where the answer is explicitly expected.

The remaining classes all involve limited bottom-up forward chaining from the utterance act. In other words, the system tries to find plans in which the utterance is a step, and then tries to find more abstract plans for which the postulated plan is a subplan, and so on. Throughout this process, postulated plans are eliminated by a set of heuristics based on those 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. When heuristics cannot eliminate all but one postulated plan, the chaining stops.

Class (3) involves not only recognizing a metaplan based on the utterance, but in satisfying its constraints, also involves connecting the metaplan to a plan on the stack. If the plan on the stack is not the top plan, the stack must be popped down to this plan before the new metaplan is added to the stack.

Class (4) may involve not only recognizing metaplans from scratch, but also recursively constructing a plausible plan for the metaplan to be about. This occurs most frequently at the start of a dialogue. This will be shown in the examples.

For all of the preference classes, once a plan or set of plans is recognized, it is expanded by adding the definitions of all steps and substeps until there is no unique expansion for any of the remaining substeps.

If there are multiple interpretations remaining at the end of this process, multiple versions of the stack are created to record each possibility. There are then several ways in which one might be chosen over the others. For example, if it is the hearer's turn in the dialogue (i.e., no

additional utterance is expected from the speaker), then the hearer must initiate a clarification subdialogue. If it is still the speaker's turn, the hearer may wait for further dialogue to distinguish between the possibilities.

### 3. Communicative Analysis and Interaction with Task Analysis

Much research in recent years has studied largely domain-independent linguistic issues. Since our work concentrates on incorporating the results of such work into our framework, rather than on a new investigation of these issues, we will first present the relevant results and then explain our work in those terms. Grosz [1977] noted that in task-oriented dialogues the task structure could be used to guide the discourse structure. She developed the notion of global focus of attention to represent the influence of the discourse structure; this proved useful for the resolution of definite noun phrases. Immediate focus [Grosz, 1977; Sidner, 1983] represented the influence of the linguistic form of the utterance and proved useful for understanding ellipsis, definite noun phrases, pronominalization, "this" and "that." Reichman [1981] developed the context space theory, in which the non-linear structure underlying a dialogue was reflected by the use of surface phenomena such as mode of reference and clue words. Clue words signaled a boundary shift between context spaces (the discourse units hierarchically structured) as well as the kind of shift, e.g., the clue word "now" indicated the start of a new context space which further developed the currently active space. However, Reichman's model was not limited to task-oriented dialogues; she accounted for a much wider range of discourse popping (e.g., topic switch), but used no task knowledge. Sacks et al. [1974] present the systematics of the turn-taking system for conversation and present the notion of adjacency pairs. That is, one way conversation is interactively governed is when speakers take turns completing such conventional, paired forms as question/answer.

Our communicative analysis is a step toward incorporating these results, with some modification, into a whole system. As in Grosz [1977], the task structure guides the focus mechanism, which marks the currently executing subtask as focused. Grosz, however, assumed an initial complete model of the task structure, as well as the mapping from an utterance to a given subtask in this

structure. Plan recognizers obviously cannot make such assumptions. Carberry [1983] provided explicit rules for tracking shifts in the task structure. From an utterance, she recognized part of the task plan, which was then used as an expectation structure for future plan recognition. For example, upon completion of a subtask, execution of the next subtask was the most salient expectation. Similarly, our focus mechanism updates the current focus by knowing what kind of plan structure traversals correspond to coherent topic continuation. These in turn provide expectations for the plan recognizer.

As in Grosz [1977] and Reichman [1981], we also use surface linguistic phenomena to help determine focus shifts. For example, clue words often explicitly mark what would be an otherwise incoherent or unexpected focus switch. Our metaplans and stack mechanism capture Reichman's manipulation of the context space hierarchies for topic suspension and resumption. Clue words become explicit markers of meta-acts. In particular, the stack manipulations can be viewed as corresponding to the following discourse situations. If the plan is already on the stack, then the speaker is continuing the current topic, or is resuming a previous (stacked) topic. If the plan is a metaplan to a stacked plan, then the speaker is commenting on the current topic, or on a previous topic that is implicitly resumed. Finally, in other cases, the speaker is introducing a new topic.

Conceptually, the communicative and task analysis work in parallel, although the parallelism is constrained by synchronization requirements. For example, when the task structure is used to guide the discourse structure [Grosz, 1977], plan recognition (production of the task structure) must be performed first. However, suppose the user suddenly changes task plans. Communicative analysis could pick up any clue words signalling this unexpected topic shift, indicating the expectation changes to the plan recognizer. What is important is that such a strategy is dynamically chosen depending on the utterance, in contrast to any a priori sequential (or even cascaded [Bolt, Beranek and Newman, Inc., 1979]) ordering. The example below illustrates the necessity of such a model of interaction.

#### 4. Example

This section illustrates the system's task and communicative processing of Dialogue 1. As above, we will concentrate on the task analysis; some discourse analysis will be briefly presented to give a feel for the complete system. We will take the role of the clerk, thus concentrating on understanding the passenger's utterances.

Currently, our system performs the plan recognition outlined here and is driven by the output of a parser using a semantic grammar for the train domain. The incorporation of the discourse mechanism is under development. The system at present does not generate natural language responses.

The following analysis of "The eight-fifty to Montreal?" is output from the parser:

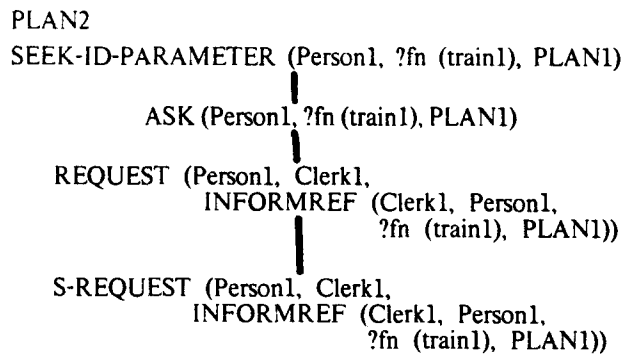
S-REQUEST (Person1, Clerk1, (R1)  
INFORMREF (Clerk1, Person1, ?fn (train1), ?plan)

with constraints: IS-PARAMETER-OF (?plan, ?fn(train1))  
arrive-station (train1) = Montreal  
depart-time (train1) = eight-fifty

In other words, Person1 is querying the clerk about some (as yet unspecified) piece of information regarding train1. In the knowledge representation, objects have a set of distinguished roles that capture their properties relevant to the domain. The notation "?fn (train1)" indicates one of these roles of train1. Throughout, the "?" notation is used to indicate skolem variables that need to be identified. S-REQUEST is a surface request, as described in Allen and Perrault [1980].

Since the stack is empty, the plan recognizer can only construct an analysis in class (4), where an entire plan stack is constructed based on the domain-specific expectations that the speaker will try to BOARD or MEET a train. From the S-REQUEST, via REQUEST, it recognizes the ASK plan and then postulates the SEEK-ID-PARAMETER plan, i.e., ASK is the only known plan for which the utterance is a step. Since its effect does not hold and its constraint is satisfied, SEEK-ID-PARAMETER can then be similarly postulated. In a more complex example, at this stage there would be competing interpretations that would need to be eliminated by the plan recognition heuristics discussed above.

In satisfying the IS-PARAMETER-OF constraint of SEEK-ID-PARAMETER, a second plan is introduced that must contain a property of a train as its parameter. This new plan will be placed on the stack before the SEEK-ID-PARAMETER plan and should satisfy one of the domain-specific expectations. An eligible domain plan is the GOTO plan, with the ?fn being either a time or a location. Since there are no plans for which SEEK-ID-PARAMETER is a step, chaining stops. The state of the stack after this plan recognition process is as follows:



CONSTRAINT: ?fn is location or time role of trains

-----
 PLAN1: GOTO (?agent, ?location, ?time)
 -----

Since SEEK-ID-PARAMETER is a metaplan, the algorithm then performs a recursive recognition on

PLAN1. This selects the BOARD plan; the MEET plan is eliminated due to constraint violation, since the arrive-station is not Toronto. Recognition of the BOARD plan also constrains ?fn to be depart-time or depart-location. The constraint on the ASK plan indicated that the speaker does not know the ?fn property of the train. Since the depart-time was known from the utterance, depart-time can be eliminated as a possibility. Thus, ?fn has been constrained to be the depart-location. Also, since the expected agent of the BOARD plan is the speaker, ?agent is set equal to Person1.

Once the recursive call is completed, plan recognition ends and all postulated plans are expanded to include the rest of their steps. The state of the stack is now as shown in Figure 3. As desired, we have constructed an entire plan stack based on the original domain-specific expectations to BOARD or MEET a train.

Recall that in parallel with the above, communicative analysis is also taking place. Once the task structure is recognized the global focus (the executing step) in each plan structure is noted. These are the S-REQUEST in the metaplan and the GOTO in the task plan. Furthermore, since R1 has been completed, the focus tracking mechanism updates the foci to the next coherent moves (the next possible steps in the task structures). These are the INFORMREF or a metaplan to the SEEK-ID-PARAMETER.

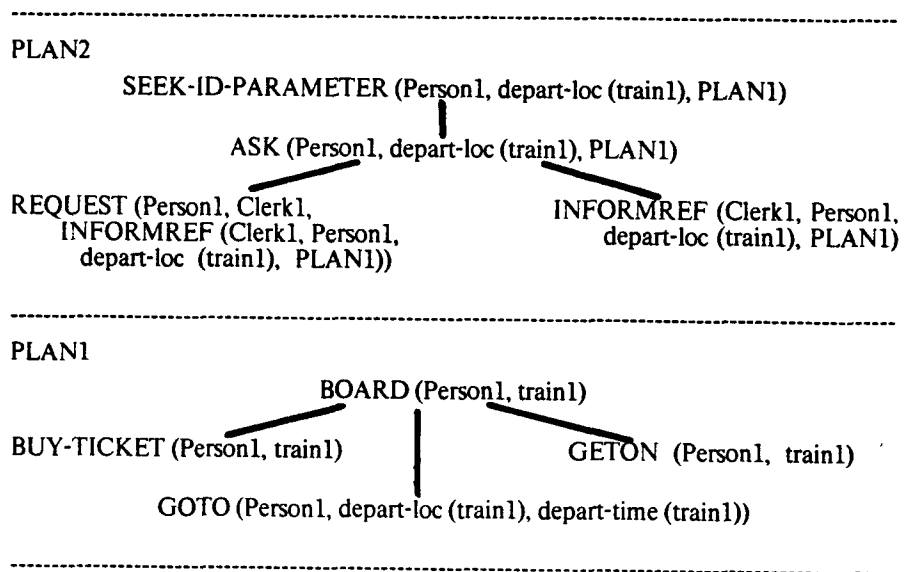


Figure 3: The Plan Stack after the First Utterance

The clerk's response to the passenger is the INFORMREF in PLAN2 as expected, which could be realized by a generation system as "Eight-fifty to Montreal. Gate seven." The global focus then corresponds to the executed INFORMREF plan step; moreover, since this step was completed the focus can be updated to the next likely task moves, a metaplan relative to the SEEK-ID-PARAMETER or a pop back to the stacked BOARD plan. Also note that this updating provides expectations for the clerk's upcoming plan recognition task.

The passenger then asks "Where is it?", i.e.,

S-REQUEST (Person1, clerk1  
INFORMREF (clerk1, Person1, loc(Gate7), ?plan)

(assuming the appropriate resolution of "it" by the immediate focus mechanism of the communicative analysis). The plan recognizer now attempts to incorporate

this utterance using the preferences described above. The first two preferences fail since the S-REQUEST does not match directly or by chaining any of the steps on the stack expected for execution. The third preference succeeds and the utterance is recognized as part of a new SEEK-ID-PARAMETER referring to the old one. This process is basically analogous to the process discussed in detail above, with the exception that the plan to which the SEEK-ID-PARAMETER refers is found in the stack rather than constructed. Also note that recognition of this metaplan satisfies one of our expectations. The other expectation involving popping the stack is not possible, for the utterance cannot be seen as a step of the BOARD plan. With the exception of the resolution of the pronoun, communicative analysis is also analogous to the above. The final results of the task and communicative analysis are shown in Figure 4. Note the inclusion of INFORM, the clerk's actual realization of the INFORMREF.

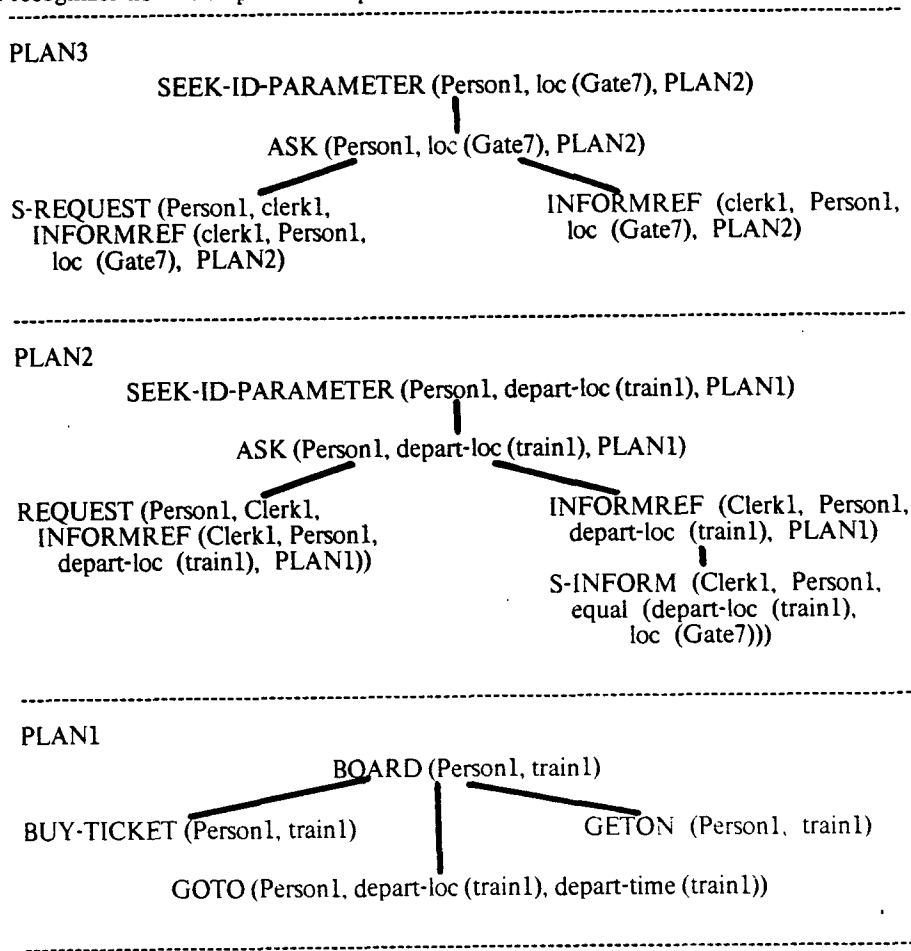


Figure 4: The Plan Stack after the Third Utterance

After the clerk replies with the INFORMREF in PLAN3, corresponding to "Down this way to the left-second one on the left," the focus updates the expected possible moves to include a metaplan to the top SEEK-ID-PARAMETER (e.g., "Second what?") or a pop. The pop allows a metaplan to the stacked SEEK-ID-PARAMETER of PLAN2 ("What's a gate?") or a pop, which allows a metaplan to the original domain plan ("It's from Toronto?"). Since the original domain plan involved no communication, there are no utterances that can be a continuation of the domain plan itself.

The dialogue concludes with the passenger's "OK. Thank you." The "OK" is an example of a clue word [Reichman, 1981], words correlated with specific manipulations to the discourse structure. In particular, "OK" may indicate a pop [Grosz, 1977], eliminating the first of the possible expectations. All but the last are then eliminated by "thank you," a discourse convention indicating termination of the dialogue. Note that unlike before, what is going on with respect to the task plan is determined via communicative analysis.

## 5. Comparisons with Other Work

### 5.1 Recognizing Speech Acts

The major difference between our present approach and previous plan recognition approaches to speech acts (e.g., [Allen and Perrault, 1980]) is that we have a hierarchy of plans, whereas all the actions in Allen and Perrault were contained in a single plan. By doing so, we have simplified the notion of what a plan is and have solved a puzzle that arose in the one-plan systems. In such systems, plans were networks of action and state descriptions linked by causality and subpart relationships, plus a set of knowledge-based relationships. This latter class could not be categorized as either a causal or a subpart relationship and so needed a special mechanism. The problem was that these relationships were not part of any plan itself, but a relationship between plans. In our system, this is explicit. The "knowref" and "know-pos" and "know-neg" relations are modeled as constraints between a plan and a metaplan, i.e., the plan to perform the task and the plan to obtain the knowledge necessary to perform the task.

Besides simplifying what counts as a plan, the multiplan approach provides some insight into how much of the user's intentions must be recognized in order to respond appropriately. We suggest that the top plan on the stack must be connected to a discourse goal. The lower plans may be only partially specified, and be filled in by later utterances. An example of this appears in considering Dialogue 2 from the first section, but there is no space to discuss this here (see [Litman and Allen, forthcoming]).

The knowledge-based relationships were crucial to the analysis of indirect speech acts (*ISA*) in Allen and Perrault [1980]. Following the argument above, this means that the indirect speech act analysis will always occur in a metaplan to the task plan. This makes sense since the ISA analysis is a communicative phenomena. As far as the task is concerned, whether a request was indirect or direct is irrelevant.

In our present system we have a set of metaplans that correspond to the common conventional ISA. These plans are abstractions of inference paths that can be derived from first principles as in Allen and Perrault. Similar "compilation" of ISA can be found in Sidner and Israel [1981] and Carberry [1983]. It is not clear in those systems, however, whether the literal interpretation of such utterances could ever be recognized. In their systems, the ISA analysis is performed before the plan recognition phase. In our system, the presence of "compiled" metaplans for ISA allows indirect forms to be considered easily, but they are just one more option to the plan recognizer. The literal interpretation is still available and will be recognized in appropriate contexts.

For example, if we set up a plan to ask about someone's knowledge (say, by an initial utterance of "I need to know where the schedule is incomplete"), then the utterance "Do you know when the Windsor train leaves?" is interpreted literally as a yes/no question because that is the interpretation explicitly expected from the analysis of the initial utterance.

Sidner and Israel [1981] outlined an approach that extended Allen and Perrault in the direction we have done as well. They allowed for multiple plans to be recognized but did not appear to relate the plans in any systematic way. Much of what we have done builds on their



suggestions and outlines specific aspects that were left unexplored in their paper. In the longer version of this paper [Litman and Allen, forthcoming], our analysis of the dialogue from their paper is shown in detail.

Grosz [1979], Levy [1979], and Appelt [1981] extended the planning framework to incorporate multiple perspectives, for example both communicative and task goal analysis; however, they did not present details for extended dialogues. ARGOT [Allen et al., 1982] was an attempt to fill this gap and led to the development of what has been presented here.

Pollack [1984] is extending plan recognition for understanding in the domain of dialogues with experts; she abandons the assumption that people always know what they really need to know in order to achieve their goals. In our work we have implicitly assumed appropriate queries and have not yet addressed this issue.

Wilensky's use of meta planning knowledge [1983] enables his planner to deal with goal interaction. For example, he has meta-goals such as resolving goal conflicts and eliminating circular goals. This treatment is similar to ours except for a matter of emphasis. His meta-knowledge is concerned with his planning mechanism, whereas our metaplans are concerned with acquiring knowledge about plans and interacting with other agents. The two approaches are also similar in that they use the same planning and recognition processes for both plans and metaplans.

## 5.2 Discourse

Although both Sidner and Israel [1981] and Carberry [1983] have extended the Allen and Perrault paradigm to deal with task plan recognition in extended dialogues, neither system currently performs any explicit discourse analysis. As described earlier, Carberry does have a (non-discourse) tracking mechanism similar to that used in [Grosz, 1977]; however, the mechanism cannot handle topic switches and resumptions, nor use surface linguistic phenomena to decrease the search space. Yet Carberry is concerned with tracking goals in an information-seeking domain, one in which a user seeks information in order to formulate a plan which will not be executed during the dialogue. (This is similar to what happens in our train domain.) Thus, her recognition procedure is also not as

tied to the task structure. Supplementing our model with metaplans provided a unifying (and cleaner) framework for understanding in both task-execution and information-seeking domains.

Reichman [1981] and Grosz [1977] used a dialogue's discourse structure and surface phenomena to mutually account for and track one another. Grosz concentrated on task-oriented dialogues with subdialogues corresponding only to subtasks. Reichman was concerned with a model underlying all discourse genres. However, although she distinguished communicative goals from speaker intent her research was not concerned with either speaker intent or any interactions. Since our system incorporates both types of analysis, we have not found it necessary to perform complex communicative goal recognition as advocated by Reichman. Knowledge of plans and metaplans, linguistic surface phenomena, and simple discourse conventions have so far sufficed. This approach appears to be more tractable than the use of rhetorical predicates advocated by Reichman and others such as Mann et al. [1977] and McKeown [1982].

Carbonell [1982] suggests that any comprehensive theory of discourse must address issues of meta-language communication, as well as integrate the results with other discourse and domain knowledge, but does not outline a specific framework. We have presented a computational model which addresses many of these issues for an important class of dialogues.

## 6. References

- Allen, J.F., A.M. Frisch, and D.J. Litman, "ARGOT: The Rochester Dialogue System," *Proc., Nat'l. Conf. on Artificial Intelligence*, Pittsburgh, PA, August 1982.
- Allen, J.F. and C.R. Perrault, "Analyzing intention in utterances," TR 50, Computer Science Dept., U. Rochester, 1979; *Artificial Intell.* 15, 3, Dec. 1980.
- Appelt, D.E., "Planning natural language utterances to satisfy multiple goals," Ph.D. thesis, Stanford U., 1981.
- Bolt, Beranek and Newman, Inc., "Research in natural language understanding," Report 4274 (Annual Report), September 1978 - August 1979.

- Carberry, S., "Tracking user goals in an information seeking environment," *Proc., Nat'l. Conf. on Artificial Intelligence*, 1983.
- Carbonell, J.G., "Meta-language utterances in purposive discourse," TR 125, Computer Science Dept., Carnegie-Mellon U., June 1982.
- Cohen, P.R. and C.R. Perrault, "Elements of a plan-based theory of speech acts," *Cognitive Science* 3, 3, 1979.
- Grosz, B.J., "The representation and use of focus in dialogue understanding," TN 151, SRI, July 1977.
- Grosz, B.J., "Utterance and objective: Issues in natural language communication," *Proc., IJCAI*, 1979.
- Horrigan, M.K., "Modelling simple dialogs," Master's Thesis, TR 108, U. Toronto, May 1977.
- Levy, D., "Communicative goals and strategies: Between discourse and syntax," in T. Givon (ed). *Syntax and Semantics* (vol. 12). New York: Academic Press, 1979.
- Litman, D.J., "Discourse and problem solving," Report 5338, Bolt Beranek and Newman, July 1983; TR 130, Computer Science Dept., U. Rochester, Sept. 1983.
- Litman, D.J. and J.F. Allen, "A plan recognition model for clarification subdialogues," forthcoming TR, Computer Science Dept., U. Rochester, expected 1984.
- Mann, W.C., J.A. Moore, and J.A. Levin, "A comprehension model for human dialogue," *Proc., 5th IJCAI*, MIT, 1977.
- McKeown, K.R., "Generating natural language text in response to questions about database structure," Ph.D. thesis, U. Pennsylvania, 1982.
- Pollack, M.E., "Goal inference in expert systems," Ph.D. thesis proposal, U. Penn., January 1984.
- Reichman, R., "Plain speaking: A theory and grammar of spontaneous discourse," Report 4681, Bolt, Beranek and Newman, Inc., 1981.
- Sacks, H., E.A. Schegloff, and G. Jefferson, "A simplest systematics for the organization of turn-taking for conversation," *Language* 50, 4, Part 1, December 1974.
- Sidner, C.L., "Focusing in the comprehension of definite anaphora," in M. Brady (ed). *Computational Models of Discourse*. Cambridge, MA: MIT Press, 1983.
- Sidner, C.L. and M. Bates, "Requirements for natural language understanding in a system with graphic displays," Report 5242, Bolt Beranek and Newman, Inc., 1983.
- Sidner, C.L. and D. Israel, "Recognizing intended meaning and speakers' plans," *Proc., 7th IJCAI*, Vancouver, B.C., August 1981.
- Wilensky, R. *Planning and Understanding*. Addison-Wesley, 1983.