UNDERSTANDING PLAN ELLIPSIS

Diane J. Litman
AT&T Bell Laboratories
3C-408A
600 Mountain Avenue
Murray Hill, NJ 07974*

ABSTRACT

This paper presents an extended and unified approach to the interpretation of sentence fragments and elliptical utterances within the context of a plan-based theory of dialogue understanding. The approach integrates knowledge about plans and knowledge about discourse, enabling the treatment of a variety of difficult linguistic phenomena within a single framework while maintaining the computational advantages of the plan-based approach.

I INTRODUCTION

Naturally occurring dialogues contain incomplete utterances difficult for existing natural language understanding systems to handle. In particular, the interpretation of many of these utterances depends not on syntactic and semantic knowledge as in most linguistic-based systems [4] [7] [9] [27] [28] but instead on pragmatic knowledge such as the underlying plans and goals of a speaker. For example, Allen [1] uses planning knowledge to interpret *sentence fragments*, syntactically incomplete utterances occurring in isolation or at the beginning of a dialogue, while Carberry [3] uses planning knowledge to interpret a class of *elliptical utterances*, syntactically incomplete utterances occurring in the course of a dialogue. This paper presents an approach that extends and unifies the interpretation of sentence fragments and elliptical utterances within the context of a plan-based theory of dialogue understanding [13]. The approach integrates knowledge about plans and knowledge about discourse, enabling the use of a single framework to handle a wide variety of difficult discourse phenomena while maintaining the computational advantages of the plan-based approach.

Consider the demands that the following dialogue (recorded at the information booth of a train station in Toronto [11]) would place on a computer system taking the role of the clerk during the understanding process.

1) Passenger: Trains going from here to Ottawa?
2) Clerk: Ottawa. Next one is at four-thirty.
3) Passenger: How about Wednesday?
4) Clerk: One at nine thirty, nine thirty in the morning, four thirty in the afternoon...yeah, that's it.

Dialogue 1

Traditional ellipsis resolution methods based on substitution into a preceding linguistic context are unable to handle the sentence fragment corresponding to utterance (1). This is because there is no linguistic context for utterance (1). Thus, the system would need to draw upon an extra-linguistic context of knowledge about the world and likely goals of the speaker. For example, the system could use the knowledge that people in train stations often want to take train trips to infer that the speaker wants to know the relevant train times for a trip to Ottawa. In other words, the system would need to recognize that the speaker's plan is to take a train trip, and recognize the utterance's relationship to this plan.

Similar points can be made with respect to the interpretation of the elliptical utterance (3). Even though a linguistic context has now been established, i.e. utterances (1) and (2), the information explicitly present in this context is still insufficient for the ellipsis resolution task. This is because the previous utterances do not contain entities that "Wednesday" can replace. Because the system could again relate the utterance to a larger context such as plans and goals, sentence fragments as well as such cases of ellipsis will be referred to as *plan ellipsis*. In this case the system could substitute into the plan underlying utterance (1) to interpret utterance (3). Furthermore, the system should be able to exploit the fact that words like "how about" often signal such utterance (and thus plan) relationships [5] [7] [8] [20] [25].

Finally, consider what the ellipsis resolution process would look like if "How about Montreal?" were to replace utterance (3). Although in this case substitution into the preceding linguistic context would suffice (with Montreal replacing Ottawa in utterance (1)), the ellipsis could alternatively be processed by again viewing utterance (3) in terms of the plan underlying utterance (1). A robust system should be able to use and coordinate linguistic and plan-based analyses of the same phenomena.

The next two sections of this paper present a plan-based framework that addresses these issues. Section II introduces the framework, followed in Section III by details needed for the plan ellipsis resolution process. Section IV illustrates the approach by tracing the processing of the dialogue above.

II PLAN RECOGNITION AND DISCOURSE ANALYSIS: AN INTEGRATED FRAMEWORK

In a plan-based approach to language understanding, an utterance is considered understood when it has been related to some underlying speaker plan in the domain of discourse. While previous works have explicitly represented and recognized such *domain plans* (e.g. take a train trip) [1] [2] [7] [24] [25], the ways that utterances could be related to such plans have been limited and not of particular concern. As a result, a variety of subdialogues as well as many forms of plan ellipsis have still proven problematic for the plan-based approach.

In the current work a set of domain-independent *discourse plans* have been introduced to explicitly represent and reason about relationships between utterances and domain plans. Technically, discourse plans refer to domain plans, i.e. they take domain plans as arguments and are thus meta-plans. Intuitively, domain plans model the contents of a topic while discourse plans model the actual manipulations of a topic. For example, there are discourse plans to introduce domain plans (topics), continue plans, specify plans, debug plans, and so on. In actuality, discourse plans can manipulate other discourse plans as well as domain plans, i.e. discourse plans can also become topics of a

*This paper describes work done at the Department of Computer Science, University of Rochester, Rochester NY 14627. It was supported in part by DARPA under Grant N00014-82-K-0193, NSF under Grant DCR8351665, and ONR under Grant N0014-80-C-0197.

conversation. In Dialogue 1 "Trains going from here to Ottawa?" achieves a discourse plan that *introduces* a domain plan to take a trip. Discussion of the domain plan is then *continued* by "Ottawa. Next one is at four-thirty" and *modified* by "How about Wednesday?" While the identification and specification of a small set of such utterance relationships has been inspired by many linguistic models of discourse [10] [17] [18] [20], the reformulation of such relationships within a plan-based framework allows their representation in terms of planning operators and their computation via a plan recognition process [14]. Section III presents detailed representations of both domain and discourse plans, while the plan recognition process [12] [13] [15] is implicitly reviewed when tracing through the example of Section IV. (Briefly, a discourse plan is recognized from every utterance via forward chaining, then a process of constraint satisfaction is used to initiate recognition of the domain and any other discourse plans related to the utterance.)

To record and monitor execution of the discourse and domain plans active at any point in a dialogue, a dialogue context in the form of a *plan stack* will also be introduced. (Many models of discourse [8] [19] [20] have argued that topic manipulations follow a stack like discipline.) During a dialogue a stack of executing and suspended plans will be built and maintained by the plan recognizer, each discourse plan referring to the plan below it, with the domain-dependent task plan on the bottom and the original discourse plan at the top. The recognition of discourse plans will be heuristically controlled by taking into account the influence of this plan stack; candidate discourse plans will be searched according to a priority order based on linguistic coherence with the current context. For example, the plan recognizer will prefer discourse plans representing stacked topic continuations to those representing topic changes.

Finally, when discourse information can be ascertained through purely linguistic means (e.g. syntactic and semantic ellipsis resolution), the information is input to the plan recognition system along with the utterance parse. Such information can then be used to either reinforce or explicitly modify the plan recognizer's processing. For example, the priority ordering of discourse plans can be overruled based on the presence of linguistic clues (e.g. phrases like "how about") correlated with less likely discourse plans. Note, however, that in the absence of such information the system can still proceed in the purely plan-based manner described above.

III DOMAIN AND DISCOURSE PLAN REPRESENTATION

In terms of the framework described above, resolution of plan ellipsis involves recognition of the domain plan underlying the elliptical utterance, and recognition of the discourse plan that actually relates the utterance to this domain plan. A hearer must thus bring to the resolution task some knowledge about typical speaker domain and discourse plans.

Schematic knowledge regarding both domain and discourse plans is represented using a standard STRIPS based notation [6] [21]. Every plan schema has a *header*, a parameterized action description that names the plan. *The parameters of a plan* are the parameters in the header. Action descriptions are defined in terms of *prerequisites*, *decompositions*, *effects* and *constraints*. Prerequisites are conditions that need to hold (or to be made to hold) before the action can be performed. Effects are conditions that will hold after the action has been successfully executed. Decompositions enable hierarchical planning. Although the action description of the header may be usefully thought of as a single action achieving a goal, such an action might not be executable. Action descriptions are in actuality composed of executable actions and possibly other action descriptions (i.e. other plans). Finally, associated with each plan is a set of applicability conditions called constraints,** which are similar to prerequisites except that the planner never attempts to achieve a constraint if it is false. The plan recognizer uses plan schemas to recognize plan instantiations underlying the production of an utterance.

Figure 1 presents a sample domain plan schema for the train station domain. The plan has header "TAKE-TRAIN-TRIP(agent, departTrain, destination)" and parameters "agent," "departTrain" and "destination," where the naming conventions for the parameters reflect an underlying type hierarchy as commonly found in semantic network systems. The plan is performed by first selecting a train, followed by buying a ticket for the train, then boarding the train. Each of these actions is itself either another action description (i.e. plan schema) or an executable action. The constraints capture the facts that the train taken, i.e. *departTrain*, goes to *destination*, that this fact is the only restriction on the set of possible candidates (*departTrainSet*) for *departTrain*, and that the ticket purchased will be used to take *departTrain*. The prerequisites and effects are not shown. Similarly the specification of other plan schemas needed in this domain, e.g. SELECT-TRAIN, BUY-TICKET, BOARD, MEET, and so on, are not shown since they will not be needed to process the example below.

```
HEADER: TAKE-TRAIN-TRIP(agent,departTrain,destination)
DECOMPOSITION:
        SELECT-TRAIN(agent, departTrain, departTrainSet)
        BUY-TICKET(agent, clerk, ticket)
        BOARD(agent, departTrain)
CONSTRAINTS:
        EQUAL(destination, arrive-station(departTrain))
        EQUAL(destination, arrive-station(departTrainSet))
        EQUAL(departTrain, object(ticket))
```

Figure 1. Domain Plan Schema for the Train Domain

Although discourse plans encode knowledge about communication, they are represented in the same way as domain plans except for the fact that they refer to other plans (i.e. they take other plans as arguments and are thus technically meta-plans). Figures 2 and 3 present several examples. The first discourse plan, MODIFY-PLAN, represents the replacement of a plan by one of several possible plan modifications. In particular, a new action is constructed from an old plan action by changing the assignment of a parameter. A modified plan is then constructed and executed by replacing the old action with this modification. The constraints specify the relationship between the plan and its modification, using a simple vocabulary for referring to and describing plans (e.g. PARAMETER, STEP). The prerequisite indicates that the plan to be modified must have already been *introduced* into the discourse context as a previous topic. (INTRODUCE-PLAN is shown in Figure 3.) The decomposition specifies that MODIFY-PLAN may be achieved by requesting execution of the modified action. As will be discussed below, all discourse plans will be recognized from *speech acts* [23] such as REQUEST. Finally, the NEXT effect states that *newAction* will be the next action performed in the modified plan, and the POP effect and REPLACE constraint explicitly overrule the normal stack behavior of the context mechanism. Informally, instead of returning to *oldPlan* upon completion of the discourse plan, the plan modification *newPlan* will instead be executed. MODIFY-PLAN is often signaled by the clue phrase "how about," as illustrated in the dialogue of Section I.

**These constraints should not be confused with the constraints of Stefik [26], which are dynamically formulated during hierarchical plan generation and represent the interactions between subproblems.

HEADER:	MODIFY-PLAN(speaker,hearer, change, changee newAction, oldAction, oldPlan, newPlan)
PREREQUISITE:	WANT(hearer, oldPlan)
DECOMPOSITION:	REQUEST(speaker, hearer, newAction)
EFFECTS:	POP(CLOSURE(oldPlan)) NEXT(newAction)
CONSTRAINTS:	PARAMETER(oldAction, changee) STEP(oldAction, oldPlan) STEP(newAction, newPlan) EQUAL(newAction, SUBST(change, changee, oldAction)) EQUAL(TYPE(change), TYPE(changee)) ¬EQUAL(change, changee) REPLACE(stack, oldStack)

Figure 2. The Discourse Relationship of Plan Modification

Figure 3 presents the other discourse plans that will be needed for the example (see [13] for a larger set). INTRODUCE-PLAN models topic introduction as well as topic change, i.e. since INTRODUCE-PLAN has no prerequisites it can occur in any discourse context. As specified via the decomposition and constraints, INTRODUCE-PLAN takes a plan of the speaker that involves the hearer and presents it to the hearer, by requesting an action that is in the plan and has the hearer as agent. The effects specify that the hearer (assumed cooperative) will adopt the joint plan as a goal, and that the action requested will be the next action performed in the plan. Just as with MODIFY-PLAN, INTRODUCE-PLAN may be signaled by the clue phrase "how about," e.g. "How about the movies?" The second plan of Figure 3, IDENTIFY-PARAMETER, models clarifications corresponding to parameter specification. In particular, by executing IDENTIFY-PARAMETER *speaker* provides *hearer* with a description of *parameter* that is informative enough to allow *hearer* to execute *action* in *plan*. As with the previous discourse plans, the decomposition is specified via a speech act and the relationships between the discourse plan and the plan being clarified are specified by the constraints.

To illustrate how these discourse plans represent the relationships between an utterance and its plan context, consider the following (slightly cleaned-up) dialogue fragment between a computer user and operator [16]:***

1)User:	Please mount a magtape for me.
2)	It's tape1.
3)Operator:	We are not allowed to mount that magtape.
4)	You will have to talk to operator2 about it.
5)User:	How about tape tape2?

Dialogue 2

The user's first utterance *introduces* a plan out of the tape domain, a plan which he or she then *clarifies* (utterance 2) and later *modifies* (utterance 5). In terms of instantiations of the schemas given above, utterances (1), (2) and (5) would be recognized as executing INTRODUCE-PLAN(user, system, mount a tape, mount plan), IDENTIFY-PARAMETER(user, system, tape, mount a tape, mount plan), and MODIFY-PLAN(user, system, tape2, tape1, mount tape2, mount tape1, tape1 mount plan, tape2 mount plan), respectively. Although new domain plans are needed to process this dialogue (e.g. mount a tape), the discourse plans and the plan recognition algorithm will remain the same across domains [13].

Finally, all that remains to be discussed are the definitions of the speech acts REQUEST and INFORMREF, used in the discourse plan decompositions given above. Basically the treatment of the speech acts is identical to the treatment given in Allen and Perrault [1]. For example, speech act decompositions are specified in terms of various *surface linguistic acts* (e.g. SURFACE-REQUEST), utterance templates correlated with sentence mood. However, to allow sentence fragments and elliptical utterances such as definite noun phrases at any point in a discourse, a new surface linguistic act called SURFACE-NP has also been included. (Carberry [3] contains a somewhat similar proposal.) As we will see, the addition of this decomposition connects incomplete utterance resolution with the plan recognition process. In particular, an incomplete utterance will be parsed as a SURFACE-NP, then the underlying speech act, discourse and domain plans recognized from the SURFACE-NP via the normal plan recognition process. Figure 4 presents the details of the SURFACE-NP addition, where CONTAINS(x, noun-phrase) means that x involves the noun-phrase as a parameter, or recursively as a parameter of a parameter, and so on. As in [1] a typical REQUEST is interrogative and a typical INFORM declarative. INFORMREF and INFORMIF are two variations of INFORM needed to handle wh-questions and yes/no questions, respectively. For example, "When does the train leave?" is a REQUEST to INFORMREF.

HEADER:	INTRODUCE-PLAN(speaker,hearer,action,plan)
DECOMPOSITION:	REQUEST(speaker, hearer, action)
EFFECTS:	WANT(hearer, plan) NEXT(action, plan)
CONSTRAINTS:	STEP(action, plan) AGENT(action, hearer)
HEADER:	IDENTIFY-PARAMETER(speaker, hearer, parameter, action, plan)
DECOMPOSITION:	INFORMREF(speaker, hearer, term, proposition)
EFFECTS:	NEXT(action, plan) KNOW(hearer, parameter, action, plan)
CONSTRAINTS:	PARAMETER(parameter, action) STEP(action, plan) PARAMETER(parameter, proposition) PARAMETER(term, proposition) WANT(hearer, plan)

Figure 3. Plan Introduction and Clarification

HEADER:	REQUEST(speaker,hearer,action)
DECOMPOSITION:	SURFACE-NP(speaker,hearer,noun-phrase)
CONSTRAINT:	CONTAINS(action,noun-phrase)
HEADER:	INFORM(speaker,hearer,proposition)
DECOMPOSITION:	SURFACE-NP(speaker,hearer,noun-phrase)
CONSTRAINT:	CONTAINS(proposition,noun-phrase)

Figure 4. Elliptical Speech Act Schemas

IV EXAMPLE

This section uses the framework developed in the last two sections to illustrate the system's processing**** of Dialogue 1.

***[13] contains a full analysis of this and several other examples

****Although the behavior to be described is fully specified by the theory, the implementation is partial and corresponds to the major contribution of the theory (i.e. the new model of plan recognition). However, all simulated computational processes have been implemented in other systems. [13] contains a full discussion of the implementation.

The example is first traced to show how information missing in fragments and elliptical utterances can be recovered as a side effect of the plan recognition process. Utterance (3) is then simplified as in Section I, illustrating the coordination of plan-based analyses with more traditional linguistic analyses when available.

A typical syntactic and semantic analysis of "Trains going from here to Ottawa?" e.g. SURFACE-NP (person1, clerk1, departTrainSet1) with EQUAL (arrive-station(departTrainSet1), Ottawa), is given as input to the plan recognition process of forward chaining. Although the SURFACE-NP matches the decompositions of both the REQUEST and INFORM schemas, since the mood is interrogative the plan recognizer prefers the REQUEST as in [1]. If the mood was declarative, the match with the decomposition of the INFORM schema would have been preferred. Furthermore, since in this particular domain the clerk's role is to provide information via speech acts, the parameter *action* of REQUEST is hypothesized to be a system INFORM (either an INFORMIF or INFORMREF), constrained to contain the noun-phrase *departTrainSet1*, e.g. INFORMREF(clerk1, person1, ?departTrainSet, EQUAL(?departTrainSet, departTrainSet1)). This INFORMREF will henceforth be called I1. As we will see, this instantiation will allow the plan recognizer to postulate a passenger discourse plan to *introduce* a system discourse plan to *clarify* a passenger domain plan to *take a train trip*. In contrast no plan interpretation will be constructed from the INFORMIF interpretation of the fragment, i.e. no chain of discourse plans to a domain plan can be constructed.

The actual plan recognition process proceeds as follows. Since at the beginning of the dialogue there is no context of plan instantiations, the system expects that the speaker will try to *introduce* a domain plan instantiation. In particular, using the INTRODUCE-PLAN schema, the REQUEST to INFORMREF hypothesized above, and the plan recognition process of forward chaining via plan decompositions, the system matches the REQUEST with the decomposition of INTRODUCE-PLAN, yielding the instantiation INTRODUCE-PLAN(person1, clerk1, I1, ?plan) (call it PLAN1), with constraints STEP(I1,?plan) and AGENT(I1, clerk1). As in [1] this hypothesis is then evaluated using a set of plan heuristics, e.g. constraints of any recognized plan must be satisfiable. To satisfy the STEP constraint a plan containing I1 will be created and arbitrarily called PLAN2. Nothing more needs to be done with respect to the second constraint, since it is already satisfied.

The system then attempts to expand PLAN2 using an analogous plan recognition process. The recognizer again uses the domain and discourse schemas and postulates that I1 of PLAN2 is the decomposition of an IDENTIFY-PARAMETER(clerk1, person1, ?parameter, ?action, ?plan). Furthermore, in satisfying the constraints on this plan, i.e.

1. PARAMETER(?parameter, ?action)
2. STEP(?action, ?plan)
3. PARAMETER (?parameter, EQUAL(?departTrainSet, departTrainSet1))
4. PARAMETER(?departTrainSet, EQUAL(?departTrainSet, departTrainSet1))
5. WANT(person1, ?plan)

a third plan is introduced (constraint 5), containing SELECT-TRAIN as a step (constraint 2). This is because SELECT-TRAIN is the only action that can contain a train set parameter (constraints 1 and 3) as described via the equality of the INFORMREF (constraint 4).

Just as PLAN2, PLAN3 then becomes input to a new plan recognition process. Using the domain plan schema of Figure 1, SELECT-TRAIN of PLAN3 is hypothesized to be the decomposition of an instantiation of TAKE-TRAIN-TRIP. Since in this case no more plans are introduced, the process of plan recognition also ends. The final hypothesis is that the passenger executed a discourse plan (PLAN1) that introduced a system discourse plan (PLAN2) to clarify a parameter in a passenger domain plan (PLAN3) to take a trip.

The various effects of all the plans are then asserted, the postulated plans are expanded top down to include the rest of their steps (based on the plan schemas), and the context mechanism pushes the plans onto the empty plan stack that represents the discourse context preceding utterance (1). Note that all three plans are recognized before any are placed on the stack. The updated stack is shown in Figure 5, with PLAN1 at the top,

PLAN1 *[completed]*

INTRODUCE-PLAN(person1, clerk1, I1, PLAN2)

REQUEST(person1, clerk1, I1)

SURFACE-NP(person1, clerk1, departTrainSet1) *[LAST]*

with EQUAL(arrive-station(departTrainSet1), Ottawa)

PLAN2

IDENTIFY-PARAMETER(clerk1,person1,departTrainSet1,S1,PLAN3)

I1:INFORMREF(clerk1,person1,?departTrainSet,
E1: EQUAL(?departTrainSet,departTrainSet1)) *[NEXT]*

PLAN3

TAKE-TRAIN-TRIP(person1, ?departTrain, Ottawa)

S1: SELECT-TRAIN (person1,?departTrain,departTrainSet1) *[NEXT]*

BUY-TICKET (person1, clerk1, ?departTicket)

BOARD (person1,?departTrain)

Figure 5. The Plan Context after Utterance (1)

PLAN2 in the middle, and PLAN3 at the bottom. In other words, the stack encodes the information that PLAN1 was executed, PLAN2 will be executed upon completion of PLAN1, and PLAN3 will be executed upon completion of PLAN2. Solid lines represent plan recognition inferences due to forward chaining, while dotted lines represent inferences due to later plan expansion. As desired, the plan recognizer has constructed a plan-based interpretation of the fragment in terms of expected discourse and domain plans, which could then be used to construct and generate a response such as the clerk's "Ottawa. Next one is at four-thirty."

Unfortunately, although the passenger is currently in the train station the train to be boarded leaves on a later date. The passenger thus uses a new utterance, "How about Wednesday?" to again try to obtain the needed information, by modifying the previous plan recognized by the system. The parser analyses "how about" as a clue phrase (using the plan recognizer's list of standard linguistic clues), "Wednesday" as SURFACE-NP (person1, clerk1, Wednesday), and inputs the information to the plan recognizer. As above the SURFACE-NP is hypothesized to be the decomposition of a REQUEST to perform some type of system INFORM involving Wednesday. Then, using the knowledge that "how about" typically signals either INTRODUCE-PLAN or MODIFY-PLAN, the plan recognizer modifies its processing. In particular, instead of assuming that the REQUEST is a topic continuation (the preferred or most coherent hypothesis in a non-null plan context), the utterance is assumed to be either a topic introduction or a topic modification. The latter hypothesis is most preferred by the coherence heuristics (MODIFY-PLAN builds on the previous context while INTRODUCE-PLAN doesn't), yielding MODIFY-PLAN (person1, clerk1, ?change, ?changee, ?action, ?oldAction, ?oldPlan, ?newPlan), where

(1) ?action is some system INFORM involving Wednesday.

The plan is then instantiated as follows. Since the REPLACE constraint indicates that MODIFY-PLAN uses an old context (here the stack of Figure 5 rather than the stack after the system's response), the prerequisite WANT(clerk1, ?oldPlan) can be satisfied by PLAN1, PLAN2 or PLAN3. Since PLAN2 is the most recently discussed, but unfinished, topic (and thus preferred via the coherence heuristics), PLAN1 is popped and the PLAN2 binding tried first. The rest of the parameters are bound via satisfaction of the following constraints:

(2) PARAMETER(?oldAction, ?changee)
(3) STEP(?oldAction, PLAN2)
(4) STEP(?action, ?newPlan)
(5) EQUAL(?action, SUBST(?change, ?changee, ?oldAction))
(6) EQUAL(TYPE(?change), TYPE(?changee))
(7) ~EQUAL(?change,?changee))

Constraint (3) can be satisfied by binding ?oldAction to I1 or IDENTIFY-PARAMETER, but with constraints (1) and (5) we know it must be bound to I1. Then, with (1), (2), (5), (6), and (7) ?action gets further specified to an INFORMREF with departTrainSet2, where EQUAL(time (departTrainSet2), Wednesday). Finally, satisfaction of constraint (4) results in the creation of a new plan (call it PLAN4) containing the new INFORMREF. This INFORMREF then becomes the input to a new plan recognition process.

As with the initial INFORMREF I1, from the new INFORMREF (call it I2) the system can recognize an IDENTIFY-PARAMETER of SELECT-TRAIN. PLAN5 is introduced to contain this SELECT-TRAIN, and another recursive recognition procedure connects SELECT-TRAIN to the higher level domain plan TAKE-TRAIN-TRIP. Note that although the first recognized plan used the previous context as a template, once the modified step was found the rest of the plan stack had to be re-recognized in order to propagate the modification. The various effects of all the plans are then asserted, in particular the effect of MODIFY-PLAN pops PLAN2 and its domain plan PLAN3 off the stack. The context mechanism then pushes the new plans on the now empty stack, as shown in Figure 6.

As a last example, consider replacing utterance (3) with the utterance "How about Montreal?" In this case the utterance is similar to a type of ellipsis handled linguistically by many existing

[COMPLETED]
MODIFY-PLAN(person1,clerk1,E2,E1,I2,I1,PLAN2,PLAN4)
REQUEST(person1,clerk1,I2)
SURFACE-NP(person1,clerk1,Wednesday) *[LAST]*

PLAN4
IDENTIFY-PARAMETER(clerk1,person1,departTrainSet2,S2,PLAN5)
I2:INFORMREF(clerk1,person1,?departTrainSet,
E2: EQUAL(?departTrainSet, departTrainSet2)) *[NEXT]*

PLAN5
TAKE-TRAIN-TRIP(person1, ?departTrain, Ottawa)
S2: SELECT-TRAIN (person1,?departTrain,departTrainSet2) *[NEXT]*
BUY-TICKET (person1,clerk1,?departTicket)
BOARD (person1,?departTrain)

with EQUAL(time(departTrainSet2),Wednesday)
EQUAL(arrive-station(departTrainSet2),Ottawa)

Figure 6. The Modified Plan Context

systems [4] [7] [9] [27] [28], since the noun-phrase "Montreal" can replace the previous lexical item "Ottawa." As mentioned in Section II, the system will receive such linguistic analyses of discourse phenomena along with the parser input, and use the analyses to constrain the plan recognition process. Even though such phenomena can alternatively be explained in plan-based terms, since linguistic resolution methods are typically simpler than plan-based methods such an approach increases the efficiency of the plan recognizer. For example, in the "How about Montreal?" case a complete parse would be input to the plan recognition system rather than just a SURFACE-NP. Although the complete parse would contain previously intermediate plan-based results, since the results are now known from the start the search and constraint satisfaction processes are much quicker, i.e. much of the work now involves plan verification rather than plan construction. The direct recognition of discourse plans through clue phrases (as opposed to their recognition through search processes) illustrates similar savings. As seen above, however, if such results are unavailable or if they do not lead to any plan interpretations, an alternative plan-based analysis of the discourse phenomena will eventually be provided.

V COMPARISONS AND CONCLUSIONS

Although many elliptical utterances can be understood by directly modifying a previous utterance [4] [7] [9] [27] [28], such approaches can not handle elliptical utterances when the missing portions refer to entities in a speaker's non-linguistic or pragmatic context. As discussed above, a pragmatic context is also needed when understanding sentence fragments (as well as when interpretating other linguistic phenomena such as model-interpretative anaphora [22]).

Allen and Perrault [1] were among the first to propose a plan-based pragmatic theory for the interpretation of sentence fragments. The theory was restricted, however, in that the plan recognition process could only deal with utterances in isolation rather than in the context of a dialogue. More recently Carberry has addressed the problem of ellipsis resolution by building on a plan recognition framework for dialogue understanding [2]. However, in order to process intersentential ellipsis the non-elliptical understanding process had to be supplemented with discourse knowledge and mechanisms [3]. Furthermore, the framework could not handle elliptical utterances that also represented topic change. This is in contrast to the current work, where knowledge about a general set of discourse plans was incorporated into a plan-based theory from the beginning, enabling the use of a single framework to recognize plans from sentence fragments, elliptical utterances, and non-elliptical utterances in a wide variety of dialogues.

ACKNOWLEDGEMENTS

Thanks to Ron Brachman, Julia Hirschberg, Brian Williams, and Bruce Ballard for comments on earlier versions of this paper.

REFERENCES

1. J. F. Allen and C. R. Perrault, "Analyzing Intention in Utterances", *Artificial Intelligence 15*, 3 (1980), 143-178.
2. S. Carberry, "Tracking User Goals in an Information-Seeking Environment", *AAAI*, Washington, D.C., August 1983, 59-63.
3. S. Carberry, "A Pragmatics-Based Approach to Understanding Intersentential Ellipsis", *ACL*, Chicago, July 1985, 188-197.
4. J. G. Carbonell and P. J. Hayes, "Recovery Strategies for Parsing Extragrammatical Language", *AJCL 9*, 3-4 (July-December 1983), 123-146.
5. R. Cohen, "A Computational Model for the Analysis of Arguments", Ph.D. Thesis and Tech. Rep. 151, University of Toronto, October 1983.
6. R. E. Fikes and N. J. Nilsson, "STRIPS: A new Approach to the Application of Theorem Proving to Problem Solving", *Artificial Intelligence 2*, 3/4 (1971), 189-208.
7. B. J. Grosz, "The Representation and Use of Focus in Dialogue Understanding", Technical Note 151, SRI, July 1977.
8. B. J. Grosz and C. L. Sidner, "Discourse Structure and the Proper Treatment of Interruptions", *IJCAI*, Los Angeles, August 1985, 832-839.
9. G. G. Hendrix, "Human Engineering for Applied Natural Language Processing", *IJCAI-77*, MIT, August 1977, 183-191.
10. J. R. Hobbs, "On the Coherence and Structure of Discourse", in *The Structure of Discourse*, Ablex Publishing Corporation, Forthcoming. Also CSLI (Stanford) Report No. CSLI-85-37, October 1985.
11. M. K. Horrigan, Modelling Simple Dialogs, May 1977.
12. D. J. Litman and J. F. Allen, "A Plan Recognition Model for Clarification Subdialogues", *Coling84*, Stanford, July 1984, 302-311.
13. D. J. Litman, *Plan Recognition and Discourse Analysis: An Integrated Approach for Understanding Dialogues*, PhD Thesis and Technical Report 170, University of Rochester, 1985.
14. D. J. Litman, "Linguistic Coherence: A Plan-Based Alternative", *ACL*, New York City, 1986.
15. D. J. Litman and J. F. Allen, "A Plan Recognition Model for Subdialogues in Conversation", *Cognitive Science*, to appear. Also University of Rochester Tech. Rep. 141, November 1984.
16. W. Mann, "Corpus of Computer Operator Transcripts", Unpublished Manuscript, ISI, 1970's.
17. W. C. Mann, "Discourse Structures for Text Generation", *Coling84*, Stanford, July 1984, 367-375.
18. K. R. McKeown, *Generating Natural Language Text in Response to Questions about Database Structure*, PhD Thesis, University of Pennsylvania, Philadelphia, 1982.
19. L. Polanyi and R. J. H. Scha, "The Syntax of Discourse", *Text (Special Issue: Formal Methods of Discourse Analysis) 3*, 3 (1983), 261-270.
20. R. Reichman, "Plain Speaking: A Theory and Grammar of Spontaneous Discourse", Report No. 4681, Bolt, Beranek and Newman, 1981.
21. E. D. Sacerdoti, *A Structure for Plans and Behavior*, Elsevier, New York, 1977.
22. I. Sag and J. Hankamer, "Toward a Theory of Anaphoric Processing", *Linguistic and Philosophy*, 1984, 325-345.
23. J. R. Searle, in *Speech Acts, an Essay in the Philosophy of Language*, Cambridge University Press, New York, 1969.
24. C. L. Sidner and D. J. Israel, "Recognizing Intended Meaning and Speakers' Plans", *IJCAI*, Vancouver, 1981, 203-208.
25. C. L. Sidner, "Plan Parsing for Intended Response Recognition in Discourse", *Computational Intelligence 1*, 1 (February 1985), 1-10.
26. M. Stefik, "Planning with Constraints (MOLGEN: Part 1)", *Artificial Intelligence 16* (1981), 111-140.
27. D. L. Waltz and B. A. Goodman, "Writing a Natural Language Data Base System", *IJCAI-77*, MIT, August 1977, 144-150.
28. R. M. Weischedel and N. K. Sondheimer, "An Improved Heuristic for Ellipsis Processing", *ACL*, Toronto, June 1982, 85-88.