LINGUISTIC COHERENCE: A PLAN-BASED ALTERNATIVE

Diane J. Litman
AT&T Bell Laboratories
3C-408A
600 Mountain Avenue
Murray Hill, NJ 07974[1]

ABSTRACT

To fully understand a sequence of utterances, one must be able to infer implicit relationships between the utterances. Although the identification of sets of utterance relationships forms the basis for many theories of discourse, the formalization and recognition of such relationships has proven to be an extremely difficult computational task.

This paper presents a plan-based approach to the representation and recognition of implicit relationships between utterances. Relationships are formulated as discourse plans, which allows their representation in terms of planning operators and their computation via a plan recognition process. By incorporating complex inferential processes relating utterances into a plan-based framework, a formalization and computability not available in the earlier works is provided.

INTRODUCTION

In order to interpret a sequence of utterances fully, one must know how the utterances *cohere;* that is, one must be able to infer implicit relationships as well as non-relationships between the utterances. Consider the following fragment, taken from a terminal transcript between a user and a computer operator (Mann [12]):

> Could you mount a magtape for me?
> It's tape 1.

Such a fragment appears coherent because it is easy to infer how the second utterance is related to the first. Contrast this with the following fragment:

> Could you mount a magtape for me?
> It's snowing like crazy.

This sequence appears much less coherent since now there is no obvious connection between the two utterances. While one could postulate some connection (e.g., the speaker's magtape contains a database of places to go skiing), more likely one would say that there is no relationship between the utterances. Furthermore, because the second utterance violates an expectation of discourse coherence (Reichman [16]. Hobbs [8], Grosz, Joshi, and Weinstein [6]), the utterance seems inappropriate since there are no linguistic clues (for example, prefacing the utterance with "incidentally") marking it as a topic change.

The identification and specification of sets of linguistic relationships between utterances[2] forms the basis for many computational models of discourse (Reichman [17], McKeown [14], Mann [13], Hobbs [8], Cohen [3]). By limiting the relationships allowed in a system and the ways in which relationships coherently interact, efficient mechanisms for understanding and generating well organized discourse can be developed. Furthermore, the approach provides a framework for explaining the use of surface linguistic phenomena such as *clue words,* words like "incidentally" that often correspond to particular relationships between utterances. Unfortunately, while these theories propose relationships that seem intuitive (e.g. "elaboration," as might be used in the first fragment above), there has been little agreement on what the set of possible relationships should be, or even if such a set can be defined. Furthermore, since the formalization of the relationships has proven to be an extremely difficult task, such theories typically have to depend on unrealistic computational processes. For example, Cohen [3] uses an oracle to recognize her "evidence" relationships. Reichman's [17] use of a set of conversational moves depends on the future development of extremely sophisticated semantics modules. Hobbs [8] acknowledges that his theory of coherence relations "may seem to be appealing to magic," since there are several places where he appeals to as yet incomplete subtheories. Finally, Mann [13] notes that his theory of rhetorical predicates is currently descriptive rather than constructive. McKeown's [14] implemented system of rhetorical predicates is a notable exception, but since her predicates have associated semantics expressed in terms of a specific data base system the approach is not particularly general.

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

[2] Although in some theories relationships hold between group of utterances, in others between clauses of an utterance, these distinctions will not be crucial for the purposes of this paper.

This paper presents a new model for representing and recognizing implicit relationships between utterances. Underlying linguistic relationships are formulated as discourse plans in a plan-based theory of dialogue understanding. This allows the specification and formalization of the relationships within a computational framework, and enables a plan recognition algorithm to provide the link from the processing of actual input to the recognition of underlying discourse plans. Moreover, once a plan recognition system incorporates knowledge of linguistic relationships, it can then use the correlations between linguistic relationships and surface linguistic phenomena to guide its processing. By incorporating domain independent linguistic results into a plan recognition framework, a formalization and computability generally not available in the earlier works is provided.

The next section illustrates the discourse plan representation of domain independent knowledge about communication as knowledge about the planning process itself. A plan recognition process is then developed to recognize such plans, using linguistic clues, coherence preferences, and constraint satisfaction. Finally, a detailed example of the processing of a dialogue fragment is presented, illustrating the recognition of various types of relationships between utterances.

REPRESENTING COHERENCE USING DISCOURSE PLANS

In a plan-based approach to language understanding, an utterance is considered understood when it has been related to some underlying plan of the speaker. While previous works have explicitly represented and recognized the underlying task plans of a given domain (e.g., mount a tape) (Grosz [5], Allen and Perrault [1], Sidner and Israel [21], Carberry [2], Sidner [24]), the ways that utterances could be related to such plans were limited and not of particular concern. As a result, only dialogues exhibiting a very limited set of utterance relationships could be understood.

In this work, a set of domain-independent plans about plans (i.e. meta-plans) called *discourse plans* are introduced to explicitly represent, reason about, and generalize such relationships. Discourse plans are recognized from every utterance and represent plan introduction, plan execution, plan specification, plan debugging, plan abandonment, and so on, independently of any domain. Although discourse plans can refer to both domain plans or other discourse plans, domain plans can only be accessed and manipulated via discourse plans. For example, in the tape excerpt above "Could you mount a magtape for me?" achieves a discourse plan to *introduce* a domain plan to mount a tape. "It's tape 1" then further *specifies* this domain plan.

Except for the fact that they refer to other plans (i.e. they take other plans as arguments), the representation of discourse plans is identical to the usual representation of domain plans (Fikes and Nilsson [4], Sacerdoti [18]). Every plan has a *header*, a parameterized action description that names the plan. Action descriptions are represented as operators on a planner's world model and defined in terms of *prerequisites, decompositions*, and *effects*. Prerequisites are conditions that need to hold (or to be made to hold) in the world model before the action operator can be applied. Effects are statements that are asserted into the world model after the action has been successfully executed. Decompositions enable hierarchical planning. Although the action description of the header may be usefully thought of at one level of abstraction as a single action achieving a goal, such an action might not be executable, i.e. it might be an *abstract* as opposed to *primitive* action. Abstract actions are in actuality composed of primitive actions and possibly other abstract action descriptions (i.e. other plans). Finally, associated with each plan is a set of applicability conditions called *constraints*.[3] These are similar to prerequisites, except that the planner never attempts to achieve a constraint if it is false. The plan recognizer will use such general plan descriptions to recognize the particular plan instantiations underlying an utterance.

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)

Figure 1. INTRODUCE-PLAN.

Figures 1, 2, and 3 present examples of discourse plans (see Litman [10] for the complete set). The first discourse plan, INTRODUCE-PLAN, takes a plan of the speaker that involves the hearer and presents it to the hearer (who is assumed cooperative). The decomposition specifies a typical way to do this, via execution of the *speech act* (Searle [19]) REQUEST. The constraints use a vocabulary for referring to and describing plans and actions to specify that the only actions requested will be those that are in the plan and have the hearer as agent. Since the hearer is assumed cooperative, he or she will then adopt as a goal the

[3] These constraints should not be confused with the constraints of Stefik [25], which are dynamically formulated during hierarchical plan generation and represent the interactions between subproblems.

joint plan containing the action (i.e. the first effect). The second effect states that the action requested will be the next action performed in the introduced plan. Note that since INTRODUCE-PLAN has no prerequisites it can occur in any discourse context, i.e. it does not need to be related to previous plans. INTRODUCE-PLAN thus allows the recognition of topic changes when a previous topic is completed as well as recognition of interrupting topic changes (and when not linguistically marked as such, of incoherency) at any point in the dialogue. It also captures previously implicit knowledge that at the beginning of a dialogue an underlying plan needs to be recognized.

HEADER:	CONTINUE-PLAN(speaker, hearer, step nextstep, plan)

PREREQUISITES:	LAST(step, plan)
	WANT(hearer, plan)

DECOMPOSITION:	REQUEST(speaker, hearer, nextstep)
EFFECT:	NEXT(nextstep, plan)

CONSTRAINTS:	STEP(step, plan)
	STEP(nextstep, plan)
	AFTER(step, nextstep, plan)
	AGENT(nextstep, hearer)
	CANDO(hearer, nextstep)

Figure 2. CONTINUE-PLAN.

The discourse plan in Figure 2, CONTINUE-PLAN, takes an already introduced plan as defined by the WANT prerequisite and moves execution to the next step, where the previously executed step is marked by the predicate LAST. One way of doing this is to request the hearer to perform the step that should occur after the previously executed step, assuming of course that the step is something the hearer actually can perform. This is captured by the decomposition together with the constraints. As above, the NEXT effect then updates the portion of the plan to be executed. This discourse plan captures the previously implicit relationship of coherent topic continuation in task-oriented dialogues (without interruptions), i.e. the fact that the discourse structure follows the task structure (Grosz [5]).

Figure 3 presents CORRECT-PLAN, the last discourse plan to be discussed. CORRECT-PLAN inserts a repair step into a pre-existing plan that would otherwise fail. More specifically, CORRECT-PLAN takes a pre-existing plan having subparts that do not interact as expected during execution, and debugs the plan by adding a new goal to restore the expected interactions. The pre-existing plan has subparts *laststep* and *nextstep*, where *laststep* was supposed to enable the performance of *nextstep*, but in reality did not. The plan is corrected by adding *newstep*, which enables the performance of *nextstep* and thus of the rest of *plan*. The correction can be introduced by a REQUEST for either *nextstep* or *newstep*. When *nextstep* is requested, the hearer has to use the knowledge that *nextstep* cannot currently be performed to infer that a correction must be added to the plan. When *newstep* is requested, the speaker explicitly provides the correction. The effects and constraints capture the plan situation described above and should be self-explanatory with the exception of two new terms. MODIFIES(action2, action1) means that *action2* is a variant of *action1*, for example, the same action with different parameters or a new action achieving the still required effects. ENABLES(action1, action2) means that false prerequisites of *action2* are in the effects of *action1*. CORRECT-PLAN is an example of a topic interruption that relates to a previous topic.

HEADER:	CORRECT-PLAN(speaker, hearer, laststep, newstep, nextstep, plan)

PREREQUISITES:	WANT(hearer, plan)
	LAST(laststep, plan)

DECOMPOSITION-1:	REQUEST(speaker, hearer, newstep)
DECOMPOSITION-2:	REQUEST(speaker, hearer, nextstep)

EFFECTS:	STEP(newstep, plan)
	AFTER(laststep, newstep, plan)
	AFTER(newstep, nextstep, plan)
	NEXT(newstep, plan)

CONSTRAINTS:	STEP(laststep, plan)
	STEP(nextstep, plan)
	AFTER(laststep, nextstep, plan)
	AGENT(newstep, hearer)
	~CANDO(speaker, nextstep)
	MODIFIES(newstep, laststep)
	ENABLES(newstep, nextstep)

Figure 3. CORRECT-PLAN.

To illustrate how these discourse plans represent the relationships between utterances, consider a naturally-occurring protocol (Sidner [22]) in which a user interacts with a person simulating an editing system to manipulate network structures in a knowledge representation language:

1) User: Hi. Please show the concept Person.
2) System: Drawing...OK.
3) User: Add a role called hobby.
4) System: OK.
5) User: Make the vr be Pastime.

Assume a typical task plan in this domain is to edit a structure by accessing the structure then performing a sequence of editing actions. The user's first request thus introduces a plan to edit the concept person. Each successive user utterance continues through the plan by requesting the system to perform the various editing actions. More specifically, the first utterance would correspond to INTRODUCE-PLAN (User, System, show the concept Person, edit plan). Since one of the effects of INTRODUCE-PLAN is that the system adopts the plan, the system responds by executing the next action in the plan, i.e. by showing the concept Person. The user's next utterance can then be recognized as CONTINUE-PLAN (User, System, show the concept Person, add hobby role to Person, edit plan), and so on.

Now consider two variations of the above dialogue. For example, imagine replacing utterance (5) with the User's "No, leave more room please." In this case, since the system has anticipated the requirements of future editing actions incorrectly, the user must interrupt execution of the editing task to correct the system, i.e. CORRECT-PLAN(User, System, add hobby role to Person, compress the concept Person, next edit step, edit plan). Finally, imagine that utterance (5) is again replaced, this time with "Do you know if it's time for lunch yet?" Since eating lunch cannot be related to the previous editing plan topic, the system recognizes the utterance as a total change of topic, i.e. INTRODUCE-PLAN(User, System, System tell User if time for lunch, eat lunch plan).

RECOGNIZING DISCOURSE PLANS

This section presents a computational algorithm for the recognition of discourse plans. Recall that the previous lack of such an algorithm was in fact a major force behind the last section's plan-based formalization of the linguistic relationships. Previous work in the area of domain plan recognition (Allen and Perrault [1], Sidner and Israel [21], Carberry [2], Sidner [24]) provides a partial solution to the recognition problem. For example, since discourse plans are represented identically to domain plans, the same process of plan recognition can apply to both. In particular, every plan is recognized by an incremental process of heuristic search. From an input, the plan recognizer tries to find a plan for which the input is a step,[4] and then tries to find more abstract plans for which the postulated plan is a step, and so on. After every step of this chaining process, a set of heuristics prune the candidate plan set based on assumptions regarding rational planning behavior. For example, as in Allen and Perrault [1] candidates whose effects are already true are eliminated, since achieving these plans would produce no change in the state of the world. As in Carberry [2] and Sidner and Israel [21] the plan recognition process is also incremental; if the heuristics cannot uniquely determine an underlying plan, chaining stops.

As mentioned above, however, this is not a full solution. Since the plan recognizer is now recognizing discourse as well as domain plans from a single utterance, the set of recognition processes must be coordinated.[5] An algorithm for coordinating the recognition of domain and discourse plans from a single utterance has been presented in Litman and Allen [9,11]. In brief, the plan recognizer recognizes a discourse plan from every utterance, then uses a process of constraint satisfaction to initiate recognition of the domain and any other discourse plans related to the utterance. Furthermore, 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* is built and maintained by the plan recognizer. Various models of discourse have argued that an ideal interrupting topic structure follows a stack-like discipline (Reichman [17], Polanyi and Scha [15], Grosz and Sidner [7]). The plan recognition algorithm will be reviewed when tracing through the example of the next section.

Since discourse plans reflect linguistic relationships between utterances, the earlier work on domain plan recognition can also be augmented in several other ways. For example, the search process can be constrained by adding heuristics that prefer discourse plans corresponding to the most linguistically coherent continuations of the dialogue. More specifically, in the absence of any linguistic clues (as will be described below), the plan recognizer will prefer relationships that, in the following order:

(1) continue a previous topic (e.g. CONTINUE-PLAN)

(2) interrupt a topic for a semantically related topic (e.g. CORRECT-PLAN, other corrections and clarifications as in Litman [10])

(3) interrupt a topic for a totally unrelated topic (e.g. INTRODUCE-PLAN).

Thus, while interruptions are not generally predicted, they can be handled when they do occur. The heuristics also follow the principle of Occam's razor, since they are ordered to introduce as few new plans as possible. If within one of these preferences there are still competing interpretations, the interpretation that most corresponds to a stack discipline is preferred. For example, a continuation resuming a recently interrupted topic is preferred to continuation of a topic interrupted earlier in the conversation.

Finally, since the plan recognizer now recognizes implicit relationships between utterances, linguistic clues signaling such relationships (Grosz [5], Reichman [17], Polanyi and Scha [15], Sidner [24], Cohen [3], Grosz and Sidner [7]) should be exploitable by the plan recognition algorithm. In other words, the plan recognizer should be aware of correlations between

[4] Plan chaining can also be done via effects and prerequisites. To keep the example in the next section simple, plans have been expressed so that chaining via decompositions is sufficient.

[5] Although Wilensky [26] introduced meta-plans into a natural language system to handle a totally different issue, that of concurrent goal interaction, he does not address details of coordination.

specific words and the discourse plans they typically signal. Clues can then be used both to reinforce as well as to overrule the preference ordering given above. In fact, in the latter case clues ease the recognition of topic relationships that would otherwise be difficult (if not impossible (Cohen [3], Grosz and Sidner [7], Sidner [24])) to understand. For example, consider recognizing the topic change in the tape variation earlier, repeated below for convenience:

> Could you mount a magtape for me?
> It's snowing like crazy.

Using the coherence preferences the plan recognizer first tries to interpret the second utterance as a continuation of the plan to mount a tape, then as a related interruption of this plan, and only when these efforts fail as an unrelated change of topic. This is because a topic change is least expected in the unmarked case. Now, imagine the speaker prefacing the second utterance with a clue such as "incidentally," a word typically used to signal topic interruption. Since the plan recognizer knows that "incidentally" is a signal for an interruption, the search will not even attempt to satisfy the first preference heuristic since a signal for the second or third is explicitly present.

EXAMPLE

This section uses the discourse plan representations and plan recognition algorithm of the previous sections to illustrate the processing of the following dialogue, a slightly modified portion of a scenario (Sidner and Bates [23]) developed from the set of protocols described above:

User: Show me the generic concept called "employee."
System:OK. <system displays network>
User: No, move the concept up.
System:OK. <system redisplays network>
User: Now, make an individual employee concept whose first name is "Sam" and whose last name is "Jones."

Although the behavior to be described is fully specified by the theory, the implementation corresponds only to the new model of plan recognition. All simulated computational processes have been implemented elsewhere, however. Litman [10] contains a full discussion of the implementation.

Figure 4 presents the relevant domain plans for this domain, taken from Sidner and Israel [21] with minor modifications. ADD-DATA is a plan to add new data into a network, while EXAMINE is a plan to examine parts of a network. Both plans involve the subplan CONSIDER-ASPECT, in which the user considers some aspect of a network, for example by looking at it (the decomposition shown), listening to a description, or thinking about it.

HEADER:	ADD-DATA(user, netpiece, data, screenLocation)

DECOMPOSITION:	CONSIDER-ASPECT(user, netpiece)
	PUT(system, data, screenLocation)

HEADER:	EXAMINE(user, netpiece)
DECOMPOSITION:	CONSIDER-ASPECT(user, netpiece)
HEADER:	CONSIDER-ASPECT(user, netpiece)
DECOMPOSITION:	DISPLAY(system, user, netpiece)

Figure 4. Graphic Editor Domain Plans.

The processing begins with a speech act analysis of "Show me the generic concept called 'employee'"

REQUEST (user, system, D1:DISPLAY (system, user, E1))

where E1 stands for "the generic concept called 'employee.'" As in Allen and Perrault [1], determination of such a literal[6] speech act is fairly straightforward. Imperatives indicate REQUESTS and the propositional content (e.g. DISPLAY) is determined via the standard syntactic and semantic analysis of most parsers.

Since at the beginning of a dialogue there is no discourse context, the plan recognizer tries to introduce a plan (or plans) according to coherence preference (3). Using the plan schemas of the second section, the REQUEST above, and the process of forward chaining via plan decomposition, the system postulates that the utterance is the decomposition of INTRODUCE-PLAN(user, system, D1, ?plan), where STEP(D1, ?plan) and AGENT(D1, system). The hypothesis is then evaluated using the set of plan heuristics, e.g. the effects of the plan must not already be true and the constraints of every recognized plan must be satisfiable. To satisfy the STEP constraint a plan containing D1 will be created. Nothing more needs to be done with respect to the second constraint since it is already satisfied. Finally, since INTRODUCE-PLAN is not a step in any other plan, further chaining stops.

The system then expands the introduced plan containing D1, using an analogous plan recognition process. Since the display action could be a step of the CONSIDER-ASPECT plan, which itself could be a step of either the ADD-DATA or EXAMINE plans, the domain plan is ambiguous. Note that heuristics can not eliminate either possibility, since at the beginning of the dialogue any domain plan is a reasonable expectation. Chaining halts at this branch point and since no more plans are introduced the process of plan recognition also ends. The final hypothesis is that the

[6] See Litman [10] for a discussion of the treatment of indirect speech acts (Searle [20]).

user executed a discourse plan to introduce either the domain plan ADD-DATA or EXAMINE.

Once the plan structures are recognized, their effects are asserted and the postulated plans are expanded top down to include any other steps (using the information in the plan descriptions). The plan recognizer then constructs a stack representing each hypothesis, as shown in Figure 5. The first stack has PLAN1 at the top, PLAN2 at the bottom, and encodes the information that PLAN1 was executed while PLAN2 will be executed upon completion of PLAN1. The second stack is analogous. 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 utterance in terms of expected discourse and domain plans, an interpretation which can then be used to construct and generate a response. For example, in either hypothesis the system can pop the completed plan introduction and execute D1, the next action in both domain plans. Since the higher level plan containing D1 is still ambiguous, deciding exactly what to do is an interesting plan generation issue.

Unfortunately, the system chooses a display that does not allow room for the insertion of a new concept, leading to the user's response "No, move the concept up." The utterance is parsed and input to the plan recognizer as the clue word "no" (using the plan recognizer's list of standard linguistic clues) followed by the REQUEST(user, system, M1:MOVE(system, E1, up)) (assuming the resolution of "the concept" to E1). The plan recognition algorithm then proceeds in both contexts postulated above. Using the knowledge that "no" typically does not signal a topic continuation, the plan recognizer first modifies its default mode of processing, i.e. the assumption that the REQUEST is a CONTINUE-PLAN (preference 1) is overruled. Note, however, that even without such a linguistic clue recognition of a plan continuation would have ultimately failed, since in both stacks CONTINUE-PLAN's constraint STEP(M1, PLAN2/PLAN3) would have failed. The clue thus allows the system to reach reasonable hypotheses more efficiently, since unlikely inferences are avoided.

Proceeding with preference (2), the system postulates that either PLAN2 or PLAN3 is being corrected, i.e., a discourse plan correcting one of the stacked plans is hypothesized. Since the REQUEST matches both decompositions of CORRECT-PLAN, there are two possibilities: CORRECT-PLAN(user, system, ?laststep, M1, ?nextstep, ?plan), and CORRECT-PLAN(user, system, ?laststep, ?newstep, M1, ?plan), where the variables in each will be bound as a result of constraint and prerequisite satisfaction from application of the heuristics. For example, candidate plans are only reasonable if their prerequisites were true, i.e. (in both stacks and corrections) WANT(system, ?plan) and LAST(?laststep, ?plan). Assuming the plan was executed in the context of PLAN2 or PLAN3 (after PLAN1 or PLAN1a was popped and the DISPLAY performed), ?plan could only have been bound to PLAN2 or PLAN3, and ?laststep bound to D1. Satisfaction of the constraints eliminates the PLAN3 binding, since the constraints indicate at least two steps in the plan, while PLAN3 contains a single step described at different levels of abstraction. Satisfaction of the constraints also eliminates the second CORRECT-PLAN interpretation, since STEP(M1, PLAN2) is not true. Thus only the first correction on the first stack remains plausible, and in fact, using PLAN2 and the first correction the rest of the constraints can be satisfied. In particular, the bindings yield

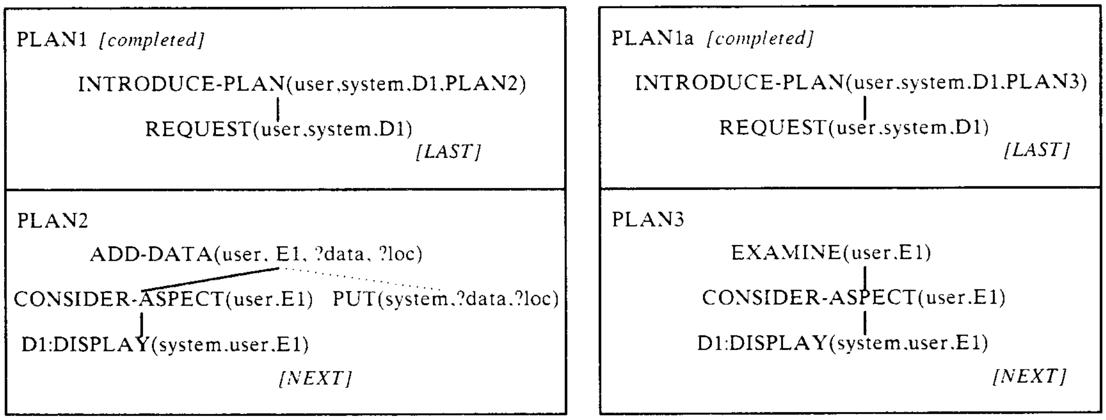

Figure 5. The Two Plan Stacks after the First Utterance.

(1) STEP(D1, PLAN2)
(2) STEP(P1, PLAN2)
(3) AFTER(D1, P1, PLAN2)
(4) AGENT(M1, system)
(5) ~CANDO(user, P1)
(6) MODIFIES(M1, D1)
(7) ENABLES(M1, P1)
where P1 stands for PUT(system, ?data, ?loc),

resulting in the hypothesis CORRECT-PLAN(user, system, D1, M1, P1, PLAN2). Note that a final possible hypothesis for the REQUEST, e.g. introduction of a new plan, is discarded since it does not tie in with any of the expectations (i.e. a preference (2) choice is preferred over a preference (3) choice).

The effects of CORRECT-PLAN are asserted (M1 is inserted into PLAN2 and marked as NEXT) and CORRECT-PLAN is pushed on to the stack suspending the plan corrected, as shown in Figure 6. The system has thus recognized not only that an interruption of ADD-DATA has occurred, but also that the relationship of interruption is one of plan correction. Note that unlike the first utterance, the plan referred to by the second utterance is found in the stack rather than constructed. Using the updated stack, the system can then pop the completed correction and resume PLAN2 with the new (next) step M1.

The system parses the user's next utterance ("Now, make an individual employee concept whose first name is 'Sam' and whose last names is 'Jones'") and again picks up an initial clue word, this time one that explicitly marks the utterance as a continuation and thus reinforces coherence preference (1). The utterance can indeed be recognized as a continuation of PLAN2, e.g. CONTINUE-PLAN(user, system, M1, MAKE1, PLAN2), analogously to the above detailed explanations. M1 and PLAN2 are bound due to prerequisite satisfaction, and MAKE1 chained through P1 due to constraint satisfaction. The updated stack is shown in Figure 7. At this stage, it would then be appropriate for the system to pop the completed CONTINUE plan and resume execution of PLAN2 by performing MAKE1.

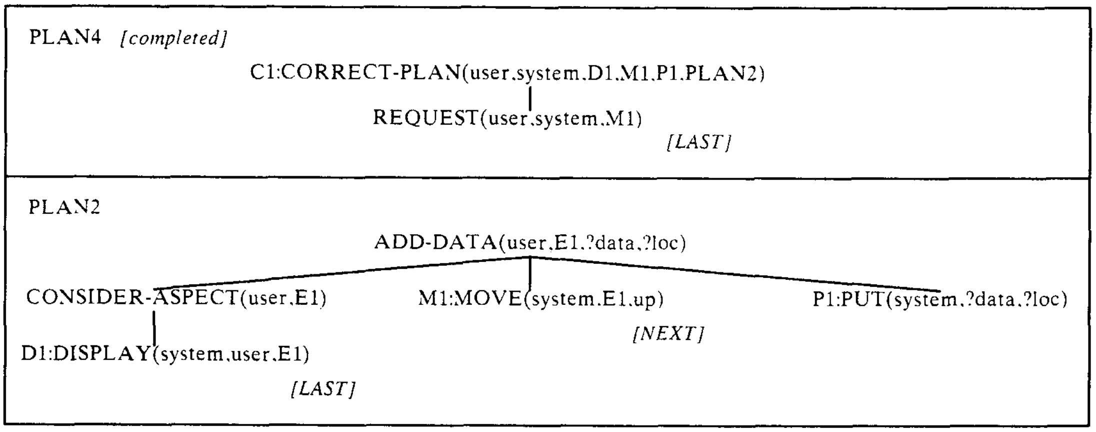
PLAN4 [completed]
C1:CORRECT-PLAN(user,system,D1,M1,P1,PLAN2)
REQUEST(user,system,M1)
[LAST]
PLAN2
ADD-DATA(user,E1,?data,?loc)
CONSIDER-ASPECT(user,E1)
M1:MOVE(system,E1,up)
P1:PUT(system,?data,?loc)
[NEXT]
D1:DISPLAY(system,user,E1)
[LAST]

Figure 6. The Plan Stack after the User's Second Utterance.

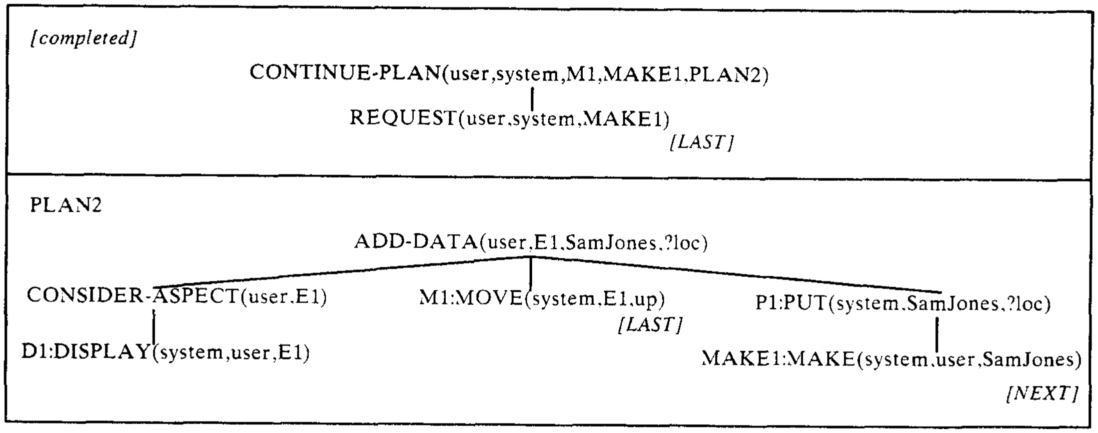
[completed]
CONTINUE-PLAN(user,system,M1,MAKE1,PLAN2)
REQUEST(user,system,MAKE1)
[LAST]
PLAN2
ADD-DATA(user,E1,SamJones,?loc)
CONSIDER-ASPECT(user,E1)
M1:MOVE(system,E1,up)
[LAST]
P1:PUT(system,SamJones,?loc)
D1:DISPLAY(system,user,E1)
MAKE1:MAKE(system,user,SamJones)
[NEXT]

Figure 7. Continuation of the Domain Plan.

CONCLUSIONS

This paper has presented a framework for both representing as well as recognizing relationships between utterances. The framework, based on the assumption that people's utterances reflect underlying plans, reformulates the complex inferential processes relating utterances within a plan-based theory of dialogue understanding. A set of meta-plans called discourse plans were introduced to explicitly formalize utterance relationships in terms of a small set of underlying plan manipulations. Unlike previous models of coherence, the representation was accompanied by a fully specified model of computation based on a process of plan recognition. Constraint satisfaction is used to coordinate the recognition of discourse plans, domain plans, and their relationships. Linguistic phenomena associated with coherence relationships are used to guide the discourse plan recognition process.

Although not the focus of this paper, the incorporation of topic relationships into a plan-based framework can also be seen as an extension of work in plan recognition. For example, Sidner [21,24] analyzed debuggings (as in the dialogue above) in terms of multiple plans underlying a single utterance. As discussed fully in Litman and Allen [11], the representation and recognition of discourse plans is a systemization and generalization of this approach. Use of even a small set of discourse plans enables the principled understanding of previously problematic classes of dialogues in several task-oriented domains. Ultimately the generality of any plan-based approach depends on the ability to represent any domain of discourse in terms of a set of underlying plans. Recent work by Grosz and Sidner [7] argues for the validity of this assumption.

ACKNOWLEDGEMENTS

I would like to thank Julia Hirschberg, Marcia Derr, Mark Jones, Mark Kahrs, and Henry Kautz for their helpful comments on drafts 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. R. Cohen, A Computational Model for the Analysis of Arguments, Ph.D. Thesis and Tech. Rep. 151, University of Toronto, October 1983.
4. 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.
5. B. J. Grosz, The Representation and Use of Focus in Dialogue Understanding, Technical Note 151, SRI, July 1977.
6. B. J. Grosz, A. K. Joshi and S. Weinstein, Providing a Unified Account of Definite Noun Phrases in Discourse, *ACL*, MIT, June 1983, 44-50.
7. B. J. Grosz and C. L. Sidner, Discourse Structure and the Proper Treatment of Interruptions, *IJCAI*, Los Angeles, August 1985, 832-839.
8. J. R. Hobbs, On the Coherence and Structure of Discourse, in *The Structure of Discourse*, L. Polanyi (ed.), Ablex Publishing Corporation, Forthcoming. Also CSLI (Stanford) Report No. CSLI-85-37, October 1985.
9. D. J. Litman and J. F. Allen, A Plan Recognition Model for Clarification Subdialogues, *Coling84*, Stanford, July 1984, 302-311.
10. D. J. Litman, *Plan Recognition and Discourse Analysis: An Integrated Approach for Understanding Dialogues*, PhD Thesis and Technical Report 170, University of Rochester, 1985.
11. 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.
12. W. Mann, Corpus of Computer Operator Transcripts, Unpublished Manuscript, ISI, 1970's.
13. W. C. Mann, Discourse Structures for Text Generation, *Coling84*, Stanford, July 1984, 367-375.
14. K. R. McKeown, *Generating Natural Language Text in Response to Questions about Database Structure*, PhD Thesis, University of Pennsylvania, Philadelphia, 1982.
15. L. Polanyi and R. J. H. Scha, The Syntax of Discourse, *Text (Special Issue: Formal Methods of Discourse Analysis) 3*, 3 (1983), 261-270.
16. R. Reichman, Conversational Coherency, *Cognitive Science 2*, 4 (1978), 283-328.
17. R. Reichman-Adar, Extended Person-Machine Interfaces, *Artificial Intelligence 22*, 2 (1984), 157-218.
18. E. D. Sacerdoti, *A Structure for Plans and Behavior*, Elsevier, New York, 1977.
19. J. R. Searle, in *Speech Acts, an Essay in the Philosophy of Language*, Cambridge University Press, New York, 1969.
20. J. R. Searle, Indirect Speech Acts, in *Speech Acts*, vol. 3, P. Cole and Morgan (ed.), Academic Press, New York, NY, 1975.

21. C. L. Sidner and D. J. Israel, Recognizing Intended Meaning and Speakers' Plans, *IJCAI*, Vancouver, 1981, 203-208.

22. C. L. Sidner, Protocols of Users Manipulating Visually Presented Information with Natural Language, Report 5128, Bolt Beranek and Newman , September 1982.

23. C. L. Sidner and M. Bates, Requirements of Natural Language Understanding in a System with Graphic Displays, Report Number 5242, Bolt Beranek and Newman Inc., March 1983.

24. C. L. Sidner, Plan Parsing for Intended Response Recognition in Discourse, *Computational Intelligence 1*, 1 (February 1985), 1-10.

25. M. Stefik, Planning with Constraints (MOLGEN: Part 1), *Artificial Intelligence 16*, (1981), 111-140.

26. R. Wilensky, *Planning and Understanding*, Addison-Wesley Publishing company, Reading, Massachusetts, 1983.