Participating in Plan-Oriented Dialogs

Alex Quilici
Artificial Intelligence Laboratory
Computer Science Department
3531 Boelter Hall
University of California
Los Angeles, CA, 90024

Abstract

Participants in plan-oriented dialogs often state beliefs about plan applicability conditions, enablements, and effects. Often, they provide these beliefs as pieces of mostly unstated chains of reasoning that justify their holding various beliefs. Understanding a dialog response requires recognizing which beliefs are being justified and inferring the unstated but necessary beliefs that are part of the justification. And producing a response requires determining which beliefs need to be justified and constructing the reasoning chains that justify holding these beliefs. This paper presents a knowledge-structure approach to these tasks. It shows how participants can use general, commonsense planning heuristics to recognize which reasoning chains are being used, and to construct the reasoning chains that justify their beliefs. Our work differs from other work on understanding dialog responses in that we focus on recognizing justifications for beliefs about a participant's plans and goals, rather than simply recognizing the plans and goals themselves. And our work differs from other work on producing dialog responses in that we rely solely on domain-independent knowledge about planning, rather than on domain- or task-specific heuristics. This approach allows us to recognize and formulate novel belief justifications.

1 Introduction

In discourse processing, two major problems are understanding the underlying connections between successive dialog responses, and producing coherent, cooperative dialog responses. In many plan-oriented dialogs, responses supply pieces of reasoning chains justifying beliefs about why one action should be executed instead of another, about whether a plan has a particular enablement or effect, and so on. In these dialogs, understanding a response involves recognizing its underlying reasoning chain and the beliefs it justifies. And, in these dialogs, producing a response involves formulating an appropriate reasoning chain for a belief in need of justification.

As an example, consider the following dialog fragment.

(1) X: The AI lab members should clean the lab.
(2) Y: But cleaning interferes with research.
(3) X: If we don't clean the lab, who will?
(4) Y: We should pay someone to clean the lab. That way we can do our research.
(5) X: Where do we get the money?
(6) Y: From the fund used to pay salaries.
(7) X: But then we can't hire as many RAs.
(8) Y: I'd rather have a clean office.
(9) X: Then why not clean it up yourself? I'd rather pay an RA than a janitor.

In each utterance X and Y present one or more beliefs, usually as part of a short chain of reasoning justifying or contradicting a belief appearing earlier in the dialog. In (1), X starts by stating a belief that lab members should clean the lab. In (2), Y responds with a belief that lab members cleaning interferes with their doing research. This belief justifies Y's unstated belief that the lab members shouldn't clean the lab. Y's underlying reasoning is that they shouldn't clean the lab because it interferes with their preferred plan of doing research. In (3), X justifies his original belief with the belief that the lab members are the only ones who can clean the lab. X's underlying reasoning is that they should clean it because there's no other way to achieve their goal of keeping it clean. And, in (4), Y states a belief that the lab members should pay someone to clean the lab, and justifies it with the belief that paying someone doesn't interfere with doing research. Y's underlying reasoning is that they should pay someone because doing so doesn't interfere with their goal of doing research, unlike their cleaning the lab themselves. The remainder of the dialog follows the same pattern. In each response, X and Y must recognize which beliefs the other is justifying and the unstated reasoning underlying that justification. And X and Y must also determine which of their beliefs need to be justified and select an appropriate set of justifying beliefs. But how can they accomplish these tasks?

This paper presents a knowledge-structure approach to participating in dialogs involving belief justifications. We show how a dialog participant can use general, common-sense planning heuristics to infer the relationship between stated beliefs and beliefs discussed earlier in the dialog, as well as to construct justifications for their beliefs.

2 Representing Dialog Responses

What types of plan-oriented beliefs appear in the responses in our example dialog? And how can these beliefs be represented?

In our example dialog there are beliefs about whether or not a particular plan should be executed, whether or not a particular plan is the best way to achieve a goal, whether or not one action is more desirable than another, whether or not two actions somehow conflict, whether or not an actor has a particular goal, and whether or not a plan has a particular enablement or effect. We represent these beliefs using *belief*(A,R), where A is the actor (either X or Y) and R is one of the planning relationships shown below in Table 1 (or their negations, which aren't shown, but are denoted as *not-*R). Here, A denotes an actor, S denotes a state (a description of properties of objects), P denotes a plan (a sequence of operators that, when executed, effects a state change), E denotes an event (an actor's execution of a particular plan), and F denotes a filler that can be either a state or an event.

Table 1: Planning relationships in our example dialog.

Relationship	*Semantics*
do(A,P)	A carries out the steps in P
causes(F,F)	F has F' as one of its effects
enables(F',F)	F is necessary for F' to occur
interferes(F,F')	F cannot occur if F does
hasgoal(A,F)	F is a goal of actor A
prefers(A,F,F')	F is more desirable than F'
applies(E,F)	E should occur to achieve F
should-do(A,E)	A should execute E

Essentially, this representation combines elements of other systems that process utterances involving plan-oriented beliefs [4, 5, 9, 15] and extends their representations to include beliefs about planning choices and preferences. More details on this representation can be found in [11], along with examples of its use to represent the plan-oriented beliefs in user/advisor dialogs. Here, however, we illustrate our representation for plan-oriented beliefs by using it to represent the first few responses in our example dialog.

In (1), X's response contains the single belief that the lab members should clean the lab.

should-do(labbies,clean)

In (2), Y states the belief that the lab members cleaning interferes with their doing research. This belief is part of a short reasoning chain that justifies Y's unstated belief that lab members shouldn't clean the lab (a belief that directly contradicts X's originally stated belief).

not-should-do(labbies,clean)
↑ *justifies*
interferes(do(labbies,clean),do(labbies,research)
prefers(labbies,do(labbies,research),do(labbies,clean))

Y's other unstated belief is that the lab members prefer doing research to cleaning. Why can X safely infer that Y holds this belief? Because if Y doesn't hold it, Y's belief that cleaning the lab interferes with doing research isn't a reasonable justification for the lab members not cleaning the lab. To justify not doing an action, that action must conflict with a more desirable action, not one that's less desirable.

In (3), X's stated belief is that no one other than the lab members can clean the lab. This belief is at the bottom of a lengthier reasoning chain.

should-do(labbies,clean)
↑ *justifies*
has-goal(labbies, keep lab clean)
applies(do(labbies, clean), keep lab clean)
↑ *justifies*
causes(do(labbies,clean),keep lab clean)
not-causes(do(other,clean),keep lab clean)

The point of X's reasoning chain is to justify the belief that the lab members should clean the lab. The justification for this belief is a pair of unstated beliefs: that the lab members want a clean lab, and that cleaning the lab is the best way to achieve this goal. This latter belief is, in turn, justified by another pair of beliefs: X's explicitly stated belief that there's no one else to keep the lab clean, and X's unstated belief that the lab members cleaning the lab results in a clean lab.

Finally, in (4), Y provides a pair of beliefs out of a slightly more complex reasoning chain.

applies(do(labbies, clean), keep lab clean)
↑ *justifies*
causes(do(labbies,pay),keep lab clean)
causes(do(labbies,clean),keep lab clean)
preferred(do(labbies,pay),do(labbies,clean)
↑ *justifies*
interferes(do(labbies,clean),do(labbies,research)
not-interferes(do(labbies,pay),do(labbies,research)
has-goal(labbies,do(labbies,research))

This reasoning chain justifies Y's stated belief that the lab members should pay someone to keep the lab clean. This belief is justified by a set of unstated beliefs: that both paying someone and cleaning the lab themselves results in a clean lab, and that the lab members prefer paying someone to cleaning. This latter belief is justified by Y's stated belief that paying someone doesn't interfere with doing research, and by Y's unstated beliefs that

the lab members have a goal of doing research, and that cleaning the lab interferes with this goal.

The remaining dialog responses have a similar representation.

This representation for dialog responses leaves X and Y with two specific tasks. First, they must infer each other's hidden reasoning chains from their stated beliefs. This involves inferring which beliefs are being justified, along with the unstated beliefs that form a necessary part of that justification. And second, they must form the reasoning chains they provide as a response. This involves pulling together a small set of justifying beliefs from their potentially large collection of domain-specific plan-oriented beliefs.

3 Representing Planning Heuristics

Our approach to inferring and forming these reasoning chains rests heavily on two assumptions. The first is that dialog participants use general, common-sense planning heuristics to justify their beliefs. The other is that these heuristics can be represented as collections of abstract planning relationships. Because these heuristics are used to justify beliefs, we call them *justification patterns*.

The first few dialog utterances use several of these heuristics. In (2), for example, Y uses one to justify a belief that the lab members shouldn't clean the lab.

> An actor shouldn't do an action if it interferes with an action the actor prefers.

This heuristic is represented as:

JP:PREVENTS-PREFERRED-ACTION

not-should-do(A,E)
↑ *justifies*
interferes(do(A,E),do(A,E'))
prefers(A,do(A,E'),do(A,E))

Y's response instantiates this JP with the actor as the lab members, the action as cleaning the lab, and the preferred action as doing research.

In (3), X uses a pair of these heuristics. X uses the first to justify the belief that the lab members should clean the lab.

> An actor should do an action if it's the most appropriate action for a goal.

This heuristic is represented as:

JP:BEST-PLAN-FOR-GOAL

should-do(A,E)
↑ *justifies*
has-goal(A,F)
applies(do(A,E),F)

X's response instantiates this JP with the actor as the lab members, the action as cleaning the lab, and the goal as keeping the lab clean. X uses the other heuristic to justify the belief that the lab members cleaning the lab is the best way to keep it clean.

> An actor should execute an action for a goal if no other actor can execute that action.

This heuristic is represented as:

JP:ONLY-ACTION-WITH-EFFECT

applies(do(A,E),F)
↑ *justifies*
causes(do(A,E),F)
not-causes(do(*other*-A,E),F)

X's response instantiates this JP with the actor as the lab members, the action as cleaning the lab, and the effect as keeping the lab clean.

Finally, in (4), Y uses two other heuristics. The first is used to justify his belief that paying someone is the best way to keep the lab clean.

> An actor should execute an action for a goal if it's preferred to an alternative that also achieves the goal.

This heuristic is represented as:

JP:PREFERRED-PLAN-WITH-EFFECT

applies(do(A,E),F)
↑ *justifies*
causes(do(A,E),F)
causes(do(A,E'),F)
prefers(A,do(A,E),do(A,E'))

Y's response instantiates this JP with the actor as the lab members, the preferred action as paying someone, and the goal as keeping the lab clean. Y uses the other heuristic to justify the belief that paying someone is preferred to cleaning the lab themselves.

> An actor prefers one action over another if it doesn't interfere with a goal and the other does.

This heuristic is represented as:

JP:OTHER-PLAN-PREVENTS-GOAL

prefers(A,do(A,E),do(A,E'))
↑ *justifies*
interferes(do(A,E'),F)
not-interferes(do(A,E),F)
has-goal(A,F)

Y's response instantiates this JP with the actor as the lab members, the preferred action as paying someone, the other action as cleaning the lab, and the goal as doing research.

Each type of planning relationship—whether a plan should be executed for an action, whether one action is preferred to another, and so on—is associated with a small set of JPs. Our example dialog contains twelve different justification patterns spanning six different planning relationships.

4 Recognizing Belief Justifications

To understand each other's responses, dialog participants must find a chain of instantiated justification pat-

terns that connect stated beliefs to beliefs that have already appeared in the dialog. Essentially, our approach is to perform a breadth-first search through the space of possible reasoning chains that begin with a stated belief and end with a belief that's directly related to the dialog (matches or negates a belief stated or inferred earlier). The premise is that a participant presents a belief either to justify one of their earlier beliefs or to contradict a belief of another participant.

We illustrate the details of the search algorithm by showing how X understands Y's response in (4). Its input is simply the set of stated beliefs. Here, that means the input is a pair of beliefs: Y's stated beliefs that the lab members should pay someone to clean the lab, and that paying someone doesn't interfere with their doing research.

applies(do(labbies,pay),keep lab clean)
not-interferes(do(labbies,pay),do(labbies,research))

The first step is to determine whether any of these stated beliefs are directly related to the dialog, and, if they are, to mark them as understood. Here, X notices that Y's first belief (that the lab members should pay someone to keep the lab clean) contradicts one of X's earlier beliefs (that the lab members should clean the lab themselves). That means X has now understood Y's first belief, which leaves X to worry only about why Y presented the other belief.

The next step is to form an initial set of potential reasoning chains by finding and instantiating JPs that can contain that belief. One relevant JP is JP:OTHER-PLAN-PREVENTS-GOAL, so X instantiates this JP, resulting in this potential reasoning chain:

prefers(labbies,do(labbies,pay),do(labbies,clean))
↑ *justifies*
interferes(do(labbies,*E'*),do(labbies,research))
not-interferes(do(labbies,pay),do(labbies,research))
has-goal(labbies,do(labbies,research))

There are also other JPs that correspond to Y's stated beliefs, and these JPs are also instantiated and added to the set of potential reasoning chains (but because of space limitations we haven't shown them here).

The next step is to match the beliefs in each of these potential reasoning chains against earlier dialog beliefs. When a match is found, the corresponding reasoning chain is further instantiated with that information. The idea is that the most likely reasoning chains are the ones partially composed of beliefs already in the dialog. In our example, X notices that one belief in the above reasoning chain matches a belief Y stated earlier. This belief, that some action interferes with doing research, corresponds to Y's earlier belief that cleaning interferes with doing research, so X instantiates the reasoning chain with this information.

prefers(*labbies*,do(labbies,pay),do(labbies,clean))
↑ *justifies*
interferes(do(labbies,clean),do(labbies,research))
not-interferes(do(labbies,pay),do(labbies,research))
has-goal(labbies,do(labbies,research))

The final step is to try to recursively connect one of these potential reasoning chains to the dialog. X checks whether any of them justify a belief that directly connects to the dialog. The belief justified by our example reasoning chain (that the lab members prefer paying someone to cleaning the lab themselves) doesn't appear in the dialog. It does, however, fall within another set of JPs. One of these is JP:PREFERRED-PLAN-WITH-EFFECT, which X then instantiates and adds to our example reasoning chain.

applies(do(labbies,pay),*F*)
↑ *justifies*
causes(do(labbies,pay),*F*)
causes(do(labbies,clean),*F*)
prefers(labbies,do(labbies,pay),do(labbies,clean))
↑ *justifies*
interferes(do(labbies,clean),do(labbies,research))
not-interferes(do(labbies,pay),do(labbies,research))
has-goal(labbies,do(labbies,research))

The belief justified by this reasoning chain (that the lab members paying someone is the best way to achieve some goal) corresponds to a belief that appeared earlier in the dialog (Y's stated belief that paying someone is the best way to keep the lab clean), so X instantiates the reasoning chain with this information.

applies(do(labbies,pay),keep lab clean)
↑ *justifies*
causes(do(labbies,pay),keep lab clean)
causes(do(labbies,clean),keep lab clean)
prefers(labbies,do(labbies,pay),do(labbies,clean))
↑ *justifies*
interferes(do(labbies,clean),do(labbies,research))
not-interferes(do(labbies,pay),do(labbies,research))
has-goal(labbies,do(labbies,research))

At this point X has found a reasoning chain that connects Y stated beliefs to the dialog.

Once a participant finds a reasoning chain that connects a stated belief to the dialog, he determines whether the beliefs it contains are shared. This is done by comparing his long-term domains-specific beliefs against the beliefs in the reasoning chain. Doing so fully instantiates the reasoning chain and brings to light any contradictory beliefs, which subsequent responses can then address. In our example, X notices that he fails to share a pair of beliefs in this reasoning chain: the belief that the lab members should pay someone to clean the lab, and the belief that the lab members paying someone results in a clean lab. That latter belief is addressed in X's next response, when he points out that there's no money to pay someone.

In the dialogs we've examined, we've observed no reasoning chain longer than four JPs, so the search terminates unsuccessfully if it can't find a connection to the dialog involving four or fewer JPs.

5 Formulating Belief Justifications

To provide a response, dialog participants must be able to form a reasoning chain justifying one of their beliefs. Our approach once again makes use of the general planning heuristics, this time performing a breadth-first search through the space of possible reasoning chains that might be used to justify a particular belief.

The search for a justification starts with a particular belief to justify. Consider, for example, how Y forms his response in (2). Y wants to justify a belief that contradicts X's belief that the lab members should clean the lab.

not-should-do(labbies,clean)

The first step is to form an initial set of possible justifying reasoning chains by instantiating all the different JPs that can justify the input belief. Here, one of those JPs is JP:PREVENTS-PREFERRED-ACTION, which, when instantiated, results in this reasoning chain.

not-should-do(labbies,clean)
↑ *justifies*
interferes(do(labbies,clean),do(labbies,*E'*))
preferred(labbies,do(labbies,*E'*),do(labbies,clean))

Other JPs for this belief are also instantiated, so there is actually a small set of possible reasoning chains (but due to space limitations we haven't shown any of the others here).

The next step is to instantiate each of these reasoning chains with beliefs already in memory. This task is accomplished by matching beliefs in the reasoning chain with beliefs already mentioned in the dialog and with long-term domain-specific beliefs. Whenever a matching belief is found, the reasoning chain is further instantiated. Our example JP suggests that Y search for a pair of beliefs: that the lab members cleaning interferes with some action, and that this action is preferred to cleaning. Y finds beliefs that cleaning conflicts with doing research, and that research is preferred to cleaning, resulting in the reasoning chain Y presents in (2).

Here, we've assumed that the matching beliefs are easily found. But it's not always possible to successfully instantiate JPs with beliefs already in memory, which means that those beliefs that aren't held in memory must be themselves justified. Consider Y's task in forming his response in (4). There Y is initially trying to justify a belief that there's a better way to keep the lab clean than for the lab members to clean it themselves.

*applies(do(labbies,*E*),keep lab clean)*

One way to justify this belief is JP:PREFERRED-PLAN-WITH-EFFECT, which Y instantiates as:

*applies(do(labbies,*E*),keep lab clean)*
↑ *justifies*
causes(do(labbies,*E*),keep lab clean)
causes(do(labbies,*E'*),keep lab clean)
prefers(labbies,do(labbies,*E*),do(labbies,*E'*))

At this point, Y searches memory for matching beliefs. Y finds one relevant belief from earlier in the dialog: cleaning the lab results in keeping the lab clean. Y instantiates the reasoning chain with that information.

applies(do(labbies,*E*),keep lab clean)
↑ *justifies*
causes(do(labbies,*E*),keep lab clean)
causes(do(labbies,clean),keep lab clean)
prefers(labbies,do(labbies,*E*),do(labbies,clean))

Y also finds a relevant belief in long-term memory: that the lab members paying someone to clean the lab results in keeping the lab clean.

applies(do(labbies,pay),keep lab clean)
↑ *justifies*
causes(do(labbies,pay),keep lab clean)
causes(do(labbies,clean),keep lab clean)
prefers(labbies,do(labbies,pay),do(labbies,clean))

This leaves one belief that Y can't find in memory: that paying someone is preferred to cleaning the lab, so Y must recursively try to justify it. The justification pattern JP:OTHER-PLAN-PREVENTS-GOAL is relevant, so Y instantiates it and adds it to the reasoning chain:

applies(do(labbies,pay),keep lab clean)
↑ *justifies*
causes(do(labbies,pay),keep lab clean)
causes(do(labbies,clean),keep lab clean)
prefers(labbies,do(labbies,pay),do(labbies,clean))
↑ *justifies*
interferes(do(labbies,clean),*F*)
not-interferes(do(labbies,pay),*F*)
has-goal(*labbies*,*F*)

Y now searches memory for matching beliefs. Y finds one relevant dialog belief: that the lab members cleaning the lab interferes with doing research. And we assume Y finds beliefs in memory that paying someone doesn't interfere with research, and that the lab members have that as a goal. Y instantiates the reasoning chain with this information, resulting in the reasoning chain presented in (4).

We actually use a bounded version of this search, limiting potential reasoning chains to a length of four. If, by that point, a dialog participant can't find a fully-grounded reasoning chain, he uses the one containing the fewest ungrounded beliefs. Our rationale is that, in practice, responses rarely provide beliefs in reasoning chains requiring more than three or four JPs to understand.

6 Current Status and Future Work

The model discussed in this paper has been implemented as part of a Prolog program. That program currently consists of two main components, a COMPREHENDER and a CONSTRUCTOR. The input to the COMPREHENDER is a representation for a set of stated participant beliefs. The output is the belief being justified by these stated beliefs, the completed reasoning chain that connects this belief to the dialog, the JPs used to form that chain, and a list of any unshared beliefs in that reasoning chain. The CONSTRUCTOR takes a belief to justify as input and uses the same general planning heuristics to formulate a justification for holding that belief. Its output is the completed reasoning chain that justifies the input belief, along with the JPs used to form that chain.

Currently, we're trying to determine how sufficient our set of justification patterns is for participating in dialogs discussing simple day-to-day planning. We're currently looking at many variants of our AI lab dialog, searching for the presence of other useful justification patterns. And we're using justification patterns for related tasks in other plan-oriented domains, such as recognizing and responding to the misconceptions of novice computer users [10, 11].

We are also working on improving our model's performance. One current problem is that the model assumes that any input belief can be connected to the dialog with a short sequence of justification patterns, and that information from later utterances isn't necessary to complete the connection. But in many dialogs a sequence of responses gradually provide the pieces of a lengthy chain of reasoning, which is impossible to understand until much of the chain is in place. This implies that competing possible justifications should be kept from utterance to utterance, and not thrown away after processing each new utterance, as our model now does. Similarly, when constructing justifications we simply throw away those reasoning chains that couldn't be completely instantiated with beliefs in memory. But it would be better to keep incompletely instantiated reasoning chains around, since the needed beliefs may appear later in the dialog. We're now revising our model to account for this phenomena.

Finally, we're considering various extensions to our model. So far our model suggests one way coherent justifications can be inferred and formed. But our model says nothing about how to compare and evaluate different justifications. And it says nothing about which pieces of the reasoning chains it forms should be presented as a response. Dialog participants, however, are capable of forming a set of coherent justifications for a beliefs and then determining which leads to the best possible response. And dialog participants also rarely present an entire reasoning chain. Instead, they selectively present one or two beliefs. Evaluating reasoning chains and determining which beliefs to present seem to require a detailed model of what beliefs dialog participants hold, beyond the beliefs that are part of the reasoning chains they provide. We're currently working on mechanisms for building this model and then reasoning on it to evaluate the newly-constructed beliefs justifications and to determine which of their beliefs to present.

7 Related Work

Our approach to recognizing and formulating dialog responses owes much to earlier research in story understanding and case-based planning [4, 15, 16, 6, 13]. These systems use high-level, abstract knowledge structures to represent and reason about the underlying theme of the story and to understand their own planning errors. Typically, these knowledge structures consist of a set of abstract planning relationships. Our approach is to use somewhat similar structures for substantially different tasks.

There are several classes of systems working on similar problems. The first consists of systems that attempt to understand the conceptual connections between dialog responses. But these systems [3, 8, 7, 14, 1] primarily focus on recognizing the participant's plan and goals and not on their beliefs about them, such as why one plan should be used instead of another.

The other class of related systems deal explicitly with belief justifications. SPIRIT [9] detects the mistaken beliefs underlying bad plans of users of a computer mail program. It infers the beliefs that led the user to a bad plan, and produces a response that points out these erroneous beliefs. Our task differs in that we're instead trying to recognize the connections between explicitly stated beliefs, and in that we're also concerned with providing responses that justify our beliefs. OpEd [2] recognizes the belief justifications present in economic editorials. But it takes a different approach, relying on linguistic clues and domain-specific reasoning, rather than on more general knowledge about planning. OpEd is also unconcerned with formulating responses that involve belief justifications. Abdul/Ilana [5] tries to participate in arguments that require belief justifications. But it uses a set of task- and domain-specific rules to recognize and understand belief justifications. And it concentrates on selecting from existing justifications for its beliefs, rather than on forming those justifications in the first place. Finally, our own earlier work [10, 11, 12], used a similar, but less general, approach to recognize and respond to the misconceptions of novice computer users. There we relied on significantly more complex justification patterns and provided no method for dealing with chains of reasoning.

8 Conclusions

Previous dialog systems have focused primarily on recognizing a participant's plans and goals. But participants don't simply present their plans and goals. Instead, they present plan-oriented beliefs as part of mostly unstated

reasoning chains that justify their beliefs. To participate in a dialog, it's necessary to recognize which reasoning chains they're using and what beliefs they're trying to justify. And it's necessary to determine which beliefs need further justification and to formulate reasoning chains that justify these beliefs. This paper has presented a knowledge-structure-based approach for accomplishing these tasks.

Our approach is attractive for several reasons. First, it builds a model of many relevant but unstated participant beliefs as a side-effect of trying to relate their utterance to the dialog. It can then reason on this model to help determine where exactly that participant went wrong. Second, it understands belief justifications using the same general, common-sense planning knowledge that it uses to formulate them. That's means the system is capable of understanding any belief justification it constructs. And finally, it shows how novel belief justifications can be understood, so long as they're formed from general planning heuristics known to the participants. That ability is especially important when involved in dialogs between participants that hold differing beliefs, since the participants can't be expected to possess all possible justifications in advance.

References

[1] J.F. Allen and C.R. Perrault, Analyzing intention in utterances. *Artificial Intelligence* 15:143–178, 1980.

[2] S. Alvarado, M. Dyer, and M. Flowers. Editorial comprehension in OpEd through argument units. In *Proceedings of the Sixth National Conference on Artificial Intelligence*. Philadelphia, PA, 1986.

[3] S. Carberry. Modeling the user's plans and goals. *Computational Linguistics*, 14(3):23–37, 1988.

[4] M.G. Dyer. *In-depth understanding: A computer model of narrative comprehension* MIT Press, Cambridge, MA, 1983.

[5] M. Flowers, R. McGuire, and L. Birnbaum. Adversary arguments and the logic of personal attacks. In *Strategies for Natural Language Processing*, Lawrence Erlbaum, Hillsdale, NJ, 1982.

[6] K. Hammond. *Case-Based Planning.* Academic Press, Cambridge, MA, 1988.

[7] H. Kautz and J.F. Allen. Generalized plan recognition. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, 423–427. Philadelphia, PA, 1986.

[8] D.J. Litman and J.F. Allen. A plan recognition model for subdialogues in conversations. *Cognitive Science*, 11:163–208, 1987.

[9] M. Pollack. A model of plan inference that distinguishes between the beliefs of actors and observers. In *Proceedings of 24th meeting of the Association of Computational Linguistics*, 207–214, New York, NY, 1986.

[10] A. Quilici. The Correction Machine: Using justification patterns to advise novice UNIX users. In *Intelligent help systems for UNIX: Case studies in Artificial Intelligence*, R. Wilensky, P. Norvig, and W. Wahlster, editors. Springer Verlag, New York, NY, 1990.

[11] A. Quilici. The Correction Machine: Formulating explanations for user misconceptions. In *Proceedings of the 11th Joint Conference on Artificial Intelligence*, Detroit, MI, 1989.

[12] A. Quilici, M.G. Dyer, and M. Flowers. Recognizing and responding to plan-oriented misconceptions. *Computational Linguistics*, 14(3):38–51, 1988.

[13] R. Schank. *Dynamic Memory.* Cambridge University Press, Cambridge, MA, 1982.

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

[15] R. Wilensky. *Planning and Understanding*, Addison Wesley, Reading, MA, 1983.

[16] R. Wilensky. Story Points. In *Strategies for Natural Language Processing.* Lawrence Erlbaum, Hillsdale, NJ, 1982.