Int. J. Human-Computer Studies (2000) **53,** 969–1015
doi:10.1006/ijhc.2000.0427
Available online at http://www.idealibrary.com on IDEAL®

Conflict resolution in collaborative planning dialogs

JENNIFER CHU-CARROLL
Lucent Technologies Bell Laboratories, 600 Mountain Avenue, Murray Hill, NJ 07974, USA. email: jencc@bell-labs.com

SANDRA CARBERRY
Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716, USA. email: carberry@cis.udel.edu

In a collaborative planning environment in which the agents are autonomous and heterogeneous, it is inevitable that discrepancies in the agents' beliefs result in conflicts during the planning process. In such cases, it is important that the agents engage in collaborative negotiation to resolve the detected conflicts in order to determine what should constitute their shared plan of actions and shared beliefs. This paper presents a plan-based model for conflict detection and resolution in collaborative planning dialogs. Our model specifies how a collaborative system should detect conflicts that arise between the system and its user during the planning process. If the detected conflicts warrant resolution, our model initiates collaborative negotiation in an attempt to resolve the conflicts in the agent's beliefs. In addition, when multiple conflicts arise, our model identifies and addresses the most effective aspect in its pursuit of conflict resolution. Furthermore, by capturing the collaborative planning process in a recursive *Propose–Evaluate–Modify* cycle of actions, our model is capable of handling embedded negotiation during conflict resolution.

1. Introduction

In collaborative planning, the participants are often autonomous and heterogeneous; thus, it is inevitable that conflicts arise among them. For example, some agents may have more extensive and accurate domain knowledge than other agents, and knowledge about the circumstances of a particular planning task may be more readily available to some agents than to others. Thus, there are discrepancies in the agents' beliefs that may result in conflicts among the agents as they collaborate on constructing a plan. In order for the collaborative planning process to proceed as smoothly as possible, the agents must be able to detect these conflicts as soon as they become evident, and attempt to resolve them in the most efficient and effective manner. As a result, the agents will engage in *collaborative negotiation subdialogs* to resolve the discrepancies in their beliefs. We follow Sidner (1994) in referring to these subdialogs as *collaborative negotiation subdialogs* because the agents are working cooperatively to reach an agreement regarding the issue in dispute. In particular, collaborative negotiation subdialogs have the following features: (1) the agents are open and honest with one another, (2) the agents do not insist on winning an argument and may change their beliefs if convincing evidence is presented to

1071-5819/00/120969 + 47 $35.00/0

them, and (3) the agents are interested in sharing beliefs with one another to determine whether their own beliefs should be revised. Such negotiation differs from argumentation (Birnbaum, Flowers & McGuire, 1980; Reichman, 1981; Cohen, 1987; Quilici, 1992; Maybury, 1993) and other kinds of negotiation, such as labor negotiation (Sycara, 1989), in that the participants are not trying to enforce their views on one another or to maximize their own benefits, but rather are trying to share their individual knowledge and beliefs in order to determine what *really* is best for the agents as a group (Chu-Carroll & Carberry, 1995*c*).

The following dialog segment, taken from a transcript of naturally occurring dialogs between travel agents and customers (SRI Transcripts, 1992), illustrates a collaborative negotiation subdialog to resolve a conflict between two agents:

(1) *C*: *I talked to C.D. again, about going from Hong Kong to Moscow.*
(2) *He made me feel like there had to be some other options besides going through Heathrow.*
(3) *T*: *Ok.*
(4) *C*: *There had to be several different cities you can go through like Beijing, to Helsinki, there has to be some options.*
(5) *T*: *Ok well, what I do is ask for connections available and that's what I'm getting is through Heathrow.*
(6) *C*: *Ok, so if somebody was in some other city and wanted to go to Moscow there has to be other options doesn't there?*
(7) *T*: *Oh I see what you mean, departing from a different city rather than Hong Kong.*
(8) *C*: *Yeah.*
(9) *T*: *Oh sure.*

In this dialog, T and C share a common goal—to construct the best plan for C to travel from Hong Kong to Moscow. T and C have previously constructed a plan in which C will travel from Hong Kong to Moscow by way of Heathrow Airport in London. However, in utterance (2), C expresses his belief that an alternative plan to traveling by way of London must exist, and in (4), provides his reasons for holding this belief. In utterance (5), T justifies her conflicting belief that going through London is the only available plan by explaining to C how this plan was obtained. In (6), C again insists on his original belief that other options exist and provides a different reason to support it. Finally in utterance (7), T confirms C's reason provided in (6) and in utterance (9) accepts C's belief that an alternative plan must exist. Thus, utterances (5)–(9) in this dialog constitute a collaborative negotiation subdialog for the purpose of squaring away the agents' conflict about whether or not an alternative to their original plan exists.

Although many researchers have developed systems that respond to user queries (McKeown, 1985; Paris, 1988; Seneff, Hirschman & Zue, 1991; Maybury, 1992; Moore & Paris, 1993; Cawsey, 1993; van Beek, Cohen & Schmidt, 1993, Logan, Reece, Cawsey, Galliers & Jones, 1994; Raskutti & Zukerman, 1994), with the exception of Logan *et al.* (1994), they either do not consider possible disagreements between the system and the user, or assume that the user will always accept the system's point of view when conflicts arise. However, our analysis of collaborative planning dialogs shows that conflicts between agents are not always resolved immediately; instead, extended collaborative negotiation subdialogs, such as the dialog segment in utterances (5)–(9), occur quite frequently. Thus, it is important that a collaborative system be able to engage in such negotiation subdialogs in a manner that will result in the conflicts being resolved naturally and efficiently.

We have implemented a plan-based system, COnflict REsolver (CORE), for conflict detection and conflict resolution during collaborative planning activities. Given a user proposal, CORE evaluates the proposal based on its private knowledge and is able to detect conflicts regarding both the validity and optimality of the proposed plan as well as conflicts about the truth of proposed beliefs. In situations where a detected conflict warrants resolution, CORE initiates negotiation subdialogs with the user to resolve the relevant conflict. In cases where multiple conflicts are detected, CORE is capable of selecting the most effective aspect to address in its pursuit of conflict resolution using its private domain beliefs and its model of the user's beliefs. Furthermore, by capturing the collaborative planning process in a recursive *Propose–Evaluate–Modify* cycle of actions, CORE is able to handle embedded negotiation during conflict resolution.

2. Modeling collaboration

2.1. CORPUS ANALYSIS

In order to detect patterns of agents' actions in collaborative planning activities, and to identify strategies that human agents employ for conflict resolution, the first author analysed sample dialogs from three corpora of collaborative planning dialogs, which are the TRAINS 91 dialogs (Gross, Allen & Traum, 1993), a set of air travel reservation dialogs (SRI Transcripts, 1992), and a set of movie/trip planning dialogs (Udel Transcripts, 1995).

These dialogs were analysed based on Sidner's model which captures collaborative planning dialogs as proposal/acceptance and proposal/rejection sequences (Sidner, 1992, 1994). Emphasis was given to situations where a proposal was not immediately accepted, indicating potential conflict between the agents. Such lack of acceptance falls into one of two categories: (1) *rejection*, where one agent rejects a proposal made by the other agent and (2) *uncertainty about acceptance*, where one agent cannot decide whether or not to accept the other agent's proposal. The former is indicated when an agent explicitly conveys rejection of the proposal and/or provides evidence that implies such rejection, while the latter is indicated when an agent solicits further information to help her decide whether or not to accept the proposal. An analysis of human strategies in response to uncertainty about acceptance of proposals is presented in Chu-Carroll and Carberry (1998), and will not be discussed further in this paper.

Our analysis confirmed both Sidner's and Walker's observations that collaborative planning dialogs can be modeled as proposal/acceptance and proposal/rejection sequences (Sidner, 1994; Walker, 1996). However, we further observed that in the vast majority of cases where a proposal is rejected, the proposal is not discarded in its entirety, but is modified to a form that will potentially be accepted by both agents. This tendency towards modification is summarized in Table 1 and is illustrated by the following example (the utterance that suggests modification of the original proposal is in boldface)

TABLE 1
Summary of corpus analysis

	No. of turns	Rejection of proposal		Focus of modification	
		Modified	Discarded	Main proposition	Child proposition
SRI	1899	39	2	32	7
TRAINS	1000	44	1	42	2
UDEL	478	45	2	29	16
Total	3377	128	5	103	25

(SRI Transcripts, 1992):

T: *The last flight out is going to be at* 1:47 *p.m. on TWA in terms of something that will get you there the same day.*
C: *mm hmm.*
T: *after that it becomes red eye flights.*
C: *ok I don't know if I want to do a red eye.* **Why don't we look at Saturday morning?**

We will use the term *collaborative negotiation* (Sidner, 1994) to refer to the kinds of negotiation employed by the human agents for conflict resolution in our transcripts. In these negotiation subdialogs, each agent is driven by the goal of devising a plan that satisfies the interests of the agents as a group, instead of one that maximizes their own individual interests. Further analysis shows that a couple of features distinguish collaborative negotiation from argumentation and non-collaborative negotiation (Chu-Carroll & Carberry, 1995*c*). First, an agent engaging in collaborative negotiation does not insist on winning an argument, and will not argue for the sake of arguing; thus, she may change her beliefs if another agent presents convincing justification for an opposing belief. This feature differentiates collaborative negotiation from argumentation (Birnbaum *et al*., 1980; Reichman, 1981; Flowers and Dyer, 1984; Cohen, 1987; Quilici, 1992). Second, agents involved in collaborative negotiation are open and honest with one another; they will not deliberately present false information to the other agents, present information in such a way as to mislead the other agents, or strategically hold back information from other agents for later use. This feature distinguishes collaborative negotiation from non-collaborative negotiation such as labor negotiation (Sycara, 1989).

As shown in Table 1, our corpus analysis also found that, in most cases where a proposal is rejected and modified, the agent directly addresses the main (and oftentimes only) proposition during conflict resolution. However, about 20% of the time, instead of directly addressing the main proposition, the agent addresses a child proposition which (1) is intended to provide support for the main proposition and (2) the agents also disagree about. Furthermore, an agent can refute either a communicated proposition or an implied evidential relationship, as illustrated by the following excerpt from a movie

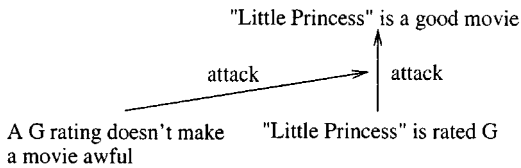


FIGURE 1. Relationship between sample utterances.

planning dialog (Udel Transcripts, 1995):

A: *"Little Princess" is supposed to be really good.*
B: *It's rated G though.*
A: *That doesn't particularly make it awful.*

Figure 1 shows the relationship between the above three utterances. The first utterance proposes the main proposition *"Little Princess" is a good movie*, about which the agents disagree. B's utterance attempts to resolve the conflict by directly addressing the main proposition and providing evidence against it. A's response to B, on the other hand, focuses on addressing the evidential relationship implicitly conveyed by B's utterance, namely that a G rating makes a movie undesirable.

2.2. THE PROPOSE–EVALUATE–MODIFY FRAMEWORK

The results of our corpus analysis suggest that when developing a computational agent that participates in collaborative planning, the following behavior should be modeled. When presented with a proposal, the agent should evaluate the proposal based on its private beliefs to determine whether to accept or reject the proposal.† If the agent rejects the proposal, instead of discarding the proposal entirely, it should attempt to modify the proposal by initiating a *collaborative negotiation subdialog* to resolve the agents' conflict about the proposal. Thus, we capture collaborative planning in a *Propose–Evaluate–Modify* cycle of actions (Chu-Carroll & Carberry, 1994, 1995*a*). This model views a collaborative planning process as agent A *proposing* a set of actions and beliefs to be added to the shared plan‡ being developed, agent B *evaluating* the proposal based on his private beliefs to determine whether or not to accept the proposal, and, if the proposal is not accepted, agent B proposing a set of *modifications* to the original proposal. Notice that this model is a recursive one in that the modification process itself contains a full collaboration cycle — agent B's proposed modifications will again be evaluated by agent A, and if conflicts arise, agent A may propose modifications to the previously proposed modifications.

†In our model, the agent can actually make a three-way decision about a proposal, to accept it, to reject it or to remain uncertain about whether the proposal should be accepted. In the last case, the agent will initiate an *information-sharing subdialog* to exchange information with the other agent so that each agent can knowledgeably re-evaluate the proposal. Further details about this process are beyond the scope of this paper, and can be found in Chu-Carroll and Carberry (1998).

‡The notion of a *shared plan* has been used in Grosz and Sidner (1990) and Allen (1991).

To illustrate how the *Propose–Evaluate–Modify* cycle of actions captures interaction between collaborative agents, consider the following dialog segment based on a transcript of naturally occurring course advisement dialogs (Columbia University Transcripts, 1985) in which an advisor (A) and a student (S) are collaborating on planning the student's schedule:

(10) *S*: *I was going to say two* [*courses*] *this time and then three next time.*
(11) *A*: *And if you take two and then don't pass one*, *you also would be slightly behind.*
(12) *S*: *Right.*
(13) *But then if I take two*, *the probability is much higher than I'll do well in both of them.*
(14) *Whereas if I take three …*
(15) *A*: *Right.*
(16) *People do take two*, *so …*

In utterance (10), S proposes a plan of taking two courses this semester and three courses next semester. A evaluates this proposal based on her private beliefs, decides that taking three courses this semester and two next semester is a better alternative than S's proposal, and in utterance (11) points out the disadvantage of S's proposal as a means of implicitly conveying her intention to modify S's proposal. S evaluates A's proposal for modification conveyed by (11), decides that although A's reasons for suggesting the alternative of taking three courses this semester and two next semester is a valid one (utterance (12)), his original proposal still constitutes a better plan; thus, in utterances (13) and (14), S provides his evidence to support his original proposal of taking two courses this semester and three next semester as an attempt to modify A's belief that taking three courses this semester is a better plan. A evaluates S's newly proposed evidence, which consists of the beliefs conveyed by utterances (13) and (14), and in (15) and (16) accepts both S's newly proposed evidence and his original proposal. Thus, our *Propose–Evaluate–Modify* cycle of actions successfully accounts for each agent's actions in negotiating the resultant plan during collaborative planning.

We argue that this *Propose–Evaluate–Modify* framework models the interaction of collaborative agents. We are interested in studying the behavior of collaborative agents who engage in dialogs with other agents in order to achieve a joint goal. Since these agents truly have an interest in satisfying the joint goal (such as making a plan for a dinner party), in most cases, they will take into consideration proposals made by their collaborators and address any disagreements instead of ignoring them. In contrast, when agents engage in debates and other types of non-collaborative activities, where the goal is to maximize each agent's own utilities (Rosenschein & Zlotkin, 1994), they may not necessarily take into account a proposal made by another agent; worse yet, they may merely discard a proposal that they consider unacceptable without even acknowledging it. Our *Propose–Evaluate–Modify* view of collaboration is further supported by the empirical studies and models of collaboration proposed in Clark and Schaefer (1989) and Clark and Wilkes-Gibbs (1990). They show that participants collaborate in maintaining a coherent discourse and that contributions in conversation involve a presentation phase and an acceptance phase. In the case of referring expressions, S1 presents a referring expression as part of an utterance; S2 then evaluates the referring expression. In the acceptance phase, S2 provides evidence that he has identified the intended entity and that it is now part of their common ground. If there are deficits in understanding, the agents enter a phase in which the referring expression is refashioned. Clark and Wilkes-Gibbs note several kinds of refashioning actions, including S1 replacing the referring expression with a new one of her own, with the intention of identifying the entity intended by S1's original expression. This notion of presentation–(evaluation)–acceptance for understanding is similar to our *Propose–Evaluate–Modify* framework for addition of actions and beliefs to the shared plan where the substitution actions in the repair phase correlate with the modification phase for conflict resolution in our framework.

In the following sections, we discuss our plan-based model for conflict detection and resolution in collaborative planning dialogs. We adopt a plan-based mechanism because it is general and easily extendible, allows the same declarative knowledge about collaborative problem-solving to be used both for inferring user intentions and for planning communicative actions, and allows the recursive nature of our model to be captured by recursive meta-plans. We present our mechanism for detecting invalid as well as suboptimal proposals, and our strategies for engaging in collaborative negotiation to resolve such conflicts. The model presented in this paper focuses on communication and negotiation between a computational agent and a human agent who are collaborating on constructing a plan to be executed by the human agent at a later point in time. The examples in this paper are taken from a university course advisement domain, the domain in which CORE has been tested, although the model can easily be applied to other domains [for examples in the air traffic control domain, see Chu-Carroll and Carberry (1996)]. Throughout this paper, the executing agent (EA) will be used to refer to the agent who will eventually be executing the plan and the system (CORE) or consulting agent (CA) will be used to refer to the computational agent who is collaborating on constructing the plan.

3. Modeling actions and intentions

In collaborative planning, the agents clearly collaborate on determining which domain actions to include in their shared plan. In the university course advisement domain, these domain actions may include agent A getting a Master's degree in CS [*Get-Masters*(*A*, *CS*)] and agent A taking a seminar course [*Take-Course(A,_seminar-course)*]. The agents will also collaborate on the strategies used to construct the domain plan. For instance, the agents may collaborate on the order in which alternative courses are considered or on whether to investigate several alternatives in parallel (Ramshaw, 1987). Furthermore, the agents may collaborate on establishing certain mutual beliefs that indirectly contribute to the construction of their domain plan. For example, the agents may collaborate on a mutual belief about whether a particular course is being offered next semester as a means of determining whether taking that course next semester is feasible. Finally, the agents engage in communicative actions in order to exchange the above desired information.

In order to capture the current intentions of the dialog participants, we use an enhanced version of the tripartite dialog model presented in Lambert and Carberry (1991). The enhanced dialog model (Chu-Carroll & Carberry, 1994) has four levels: the *domain* level which consists of the domain plan being constructed to achieve the agents' shared domain goal(s), the *problem-solving* level which contains the actions being performed to construct the domain plan, the *belief* level which consists of the mutual beliefs pursued during the planning process in order to further the problem-solving intentions, and the *discourse* level which contains the communicative actions initiated to achieve the mutual beliefs. Actions at the discourse level can contribute to other discourse actions and also establish mutual beliefs. Mutual beliefs can support other beliefs and also enable problem-solving actions. Problem-solving actions can be part of other problem-solving actions and can also enable domain actions by providing the executing agent with a plan for the domain actions.

Since the agents are engaged in *collaborative* planning, an action proposed by one agent cannot become part of their shared or joint plan until it has been accepted by the other agent. Therefore, we distinguish between the shared beliefs and actions that have been agreed upon by the participants and the newly proposed beliefs and actions about which the agents may disagree. When domain and problem-solving actions first enter the dialog model, they represent actions *proposed for execution*, while newly entered mutual beliefs represent beliefs *proposed to be held* jointly by the agents. However, discourse actions that are entered into the dialog model are *currently being executed* instead of proposed for execution; thus, the agents cannot disagree about whether to perform a discourse action, but only about whether to accept the mutual beliefs proposed by these actions (Chu-Carroll & Carberry, 1994).† Therefore, we separate the domain, problem-solving and belief levels of our dialog model into an *existing model* and a set of *proposed additions*, following Allen and Traum who differentiated among private, proposed and shared beliefs (Allen, 1991; Traum, 1993).

For instance, suppose that previous discourse has indicated that EA has the goal of getting a Bachelor of Arts degree *(Get-Bach-Arts(EA))*; Figure 2 illustrates the dialog model that will be constructed after the following utterances:

(17) *EA*: *I want to satisfy my foreign language requirement.*
(18) *Where is the exemption form for French101?*

As shown in the dialog model, EA's utterances constitute a proposal that is intended to affect the agents' shared model of domain and problem-solving intentions, as well as their mutual beliefs. Such a proposal may be explicitly stated in an utterance, such as utterance (17) explicitly proposing the domain action *Satisfy-Foreign-Language (EA)*, or implicitly conveyed by an utterance, such as utterance (18) implicitly proposing the domain action *Obtain-Exemption(EA,French101)*. Furthermore, in the dialog model, the actions and beliefs proposed by the new utterances are distinguished from those that have been accepted by both agents. The dialog model indicates that utterance (17) proposes the mutual belief that EA wants to satisfy his foreign language requirement, which is a precondition for the problem-solving action of building a plan to satisfy EA's foreign language requirement. The goal of this problem-solving action is that EA have a plan for satisfying his foreign language requirement, and having such a plan is a precondition for the domain action of EA satisfying his foreign language requirement as part of getting a Bachelor of Arts degree. On the other hand, utterance (18) proposes, at the belief level,

†We assume that all utterances have been correctly interpreted, and do not consider cases in which misunderstandings occur.

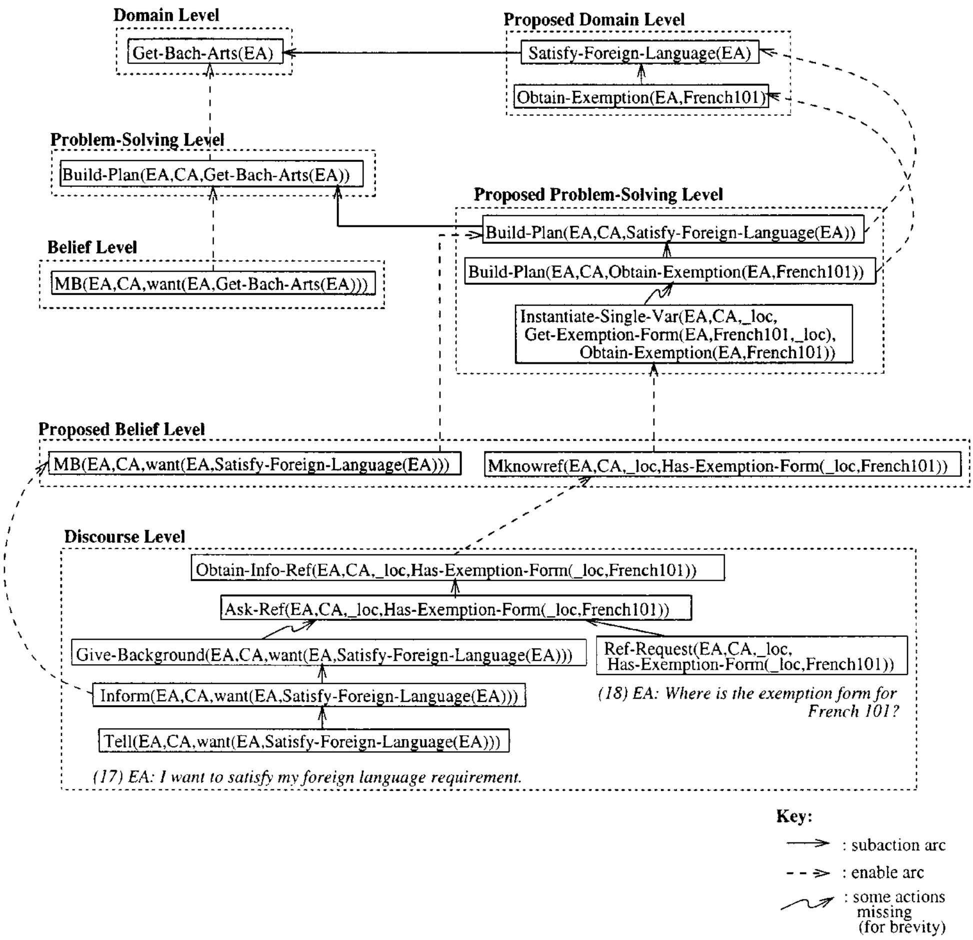


FIGURE 2. Dialog model for utterances (17) and (18).

that both agents come to know the referent of the location of the exemption form for French101. This belief is a prerequisite for the problem-solving action of instantiating the variable _loc in the subaction *Get-Exemption-Form*, which is part of building a plan for obtaining an exemption for French101. Having such a plan is again a prerequisite for executing the domain action of obtaining an exemption for French101 as part of satisfying the foreign language requirement. Thus, the question in utterance (18) implicitly proposes a mutual belief and a chain of actions, including the domain action *Obtain-Exemption(EA,French101)*. The belief and chain of inferred actions explain *why* the executing agent asked the question and suggest how the actions may be assimilated into the system's current beliefs about the agents' shared model of actions and beliefs. Notice that the actions and beliefs inferred from utterances (17) and (18) are treated as a set of proposed additions to the existing model consisting of the shared plan and shared beliefs already established between the agents.

The dialog model can be constructed incrementally from the agents' utterances via a plan recognition algorithm. In order to model proposal evaluation and conflict resolution, we hypothesize a recognition algorithm capable of recognizing intentions that (in the view of the recognizing agent) do not comprise a correct means of achieving one's goals. This is based on Lambert and Carberry's plan recognition algorithm (Lambert & Carberry, 1991), augmented to allow the system to ascribe to the executing agent erroneous beliefs that it hypothesizes the executing agent may reasonably hold. This builds on research on recognizing ill-formed plans (Pollack, 1986) and work on a relaxation algorithm for flexible plan recognition in dialog (Eller & Carberry, 1992). The proposal evaluation and modification processes discussed in this paper operate on the dialog model that would be generated by such a hypothesized recognition algorithm given a set of user utterances. However, since the focus of this paper is on response generation for conflict resolution, the recognition algorithm will not be discussed further.

4. Proposal evaluation for conflict detection

Instead of slavishly† responding to questions, a collaborative agent presented with a proposal [such as the set of proposed additions conveyed by utterances (17) and (18)] needs to decide whether or not she will accept the proposal and make it part of the shared plan being developed by the agents (Sidner, 1994). The evaluation of a proposal is carried out in two parts, the evaluation of proposed actions and the evaluation of proposed beliefs. First, the agent must decide whether or not she believes that the proposed domain and problem-solving actions will contribute to a valid plan (Pollack, 1986) as well as produce a reasonably efficient way to achieve the agents' high-level goal (Joshi, Webber & Weischedel, 1984; van Beek, 1987). The validity of a set of proposed actions is evaluated based on an agent's beliefs about recipes for performing actions, as well as her beliefs about the characteristics of the other agent. The optimality of a proposed plan, on the other hand, is evaluated with respect to an agent's beliefs about the other agent's preferences associated with the proposed actions.

The second aspect of proposal evaluation consists of the agent evaluating the proposed beliefs and determining whether or not she accepts these beliefs based on her existing private beliefs. The evaluation of proposed beliefs reasons with how multiple pieces of beliefs may form a piece of evidence that can be used to support or attack another belief, and how the effect of multiple pieces of evidence may be combined to arrive at an overall decision to accept or reject a proposed belief. Our recursive algorithm for belief evaluation takes into account the agent's private beliefs, the set of proposed beliefs, as well as the relationships among these beliefs, to annotate each belief and evidential relationship in the proposal with a decision of acceptance or rejection.

Note that our evaluation process evaluates only the proposed domain and problem-solving actions, as well as the proposed mutual beliefs. The reason that discourse actions are not evaluated is because they are being executed as the sentences are uttered; thus, the agents cannot disagree about the actions *per se*, but can only disagree about the

†Grosz and Sidner (1990) argued that a master-slave relationship does not exist among dialog participants.

implications of the discourse actions, which are represented as proposed mutual beliefs, proposed problem-solving actions and/or proposed domain actions (Chu-Carroll & Carberry, 1994).† The remainder of this section describes in turn our strategies for conflict detection with respect to proposed actions and beliefs in user proposals.

4.1. EVALUATING PROPOSED DOMAIN AND PROBLEM-SOLVING ACTIONS

As illustrated in Figure 2, a proposal at the domain or problem-solving level consists of a chain of actions inferred from an agent's utterances, in which each child action contributes to performing its parent action. Action A *contributes* to action B if the goal of action A satisfies a precondition for performing B or if action A is a subaction of action B (Grosz & Sidner, 1990). In our model, the evaluation of these proposed actions is a top–down process that detects invalid as well as suboptimal plans. The processes for detecting invalid proposals and suboptimal proposals are interleaved because we intend for the system to address the highest-level action that the agents disagree about. This is because it is meaningless to suggest, for example, a better alternative to an action when one believes that its parent action is infeasible.

4.1.1. Detecting invalid plans. Pollack argued that a plan can be invalid because one of its actions is *infeasible* or because the plan itself is *ill-formed* (Pollack, 1986). An action is infeasible if it cannot be performed by its agent(s); thus, our evaluator performs a *feasibility* check by examining whether the applicability conditions of the recipe for performing a proposed action are satisfied and whether its preconditions can be satisfied. Our evaluator considers a precondition satisfiable if there exists an action that achieves the precondition and whose applicability conditions are satisfied. Thus, only a cursory evaluation of feasibility is pursued at this stage of the planning process, with further details considered as the plan is worked out in detail. A plan is considered ill-formed if child actions do not contribute to their parent action as intended; hence, the evaluator performs a *well-formedness* check to examine, for each pair of parent–child actions in the proposal, whether the *contributes* relationship holds between them. For each action, the evaluator first performs the well-formedness check; if the contributes relation between the action and its parent (if any) is invalid, the feasibility check is ignored. This is because if an action does not contribute to performing its parent action, it cannot play its intended role in the agents' shared plan and thus its feasibility need not be considered.

Example of detecting invalid proposals: To illustrate the evaluation of proposed actions, we return to the example depicted in the dialog model in Figure 2. CORE evaluates the proposal beginning with the proposed domain actions. Since CORE believes that *Satisfy-Foreign-Language(EA)* contributes to its parent action *Get-Bach-Arts(EA)* and that *Satisfy-Foreign-Language(EA)* is feasible, CORE evaluates its child action *Obtain-Exemption(EA,French101)*. CORE believes that *Obtain-Exemption(EA,French101)* contributes to *Satisfy-Foreign-Language(EA)*; however, its recipe library indicates that an

†Although it is possible for the hearer to question the *appropriateness* of a speaker's discourse action, e.g. by saying "*why are you telling me that?*" after an *Inform* discourse action, we contend that in such cases, the agents do not disagree about the fact that an *Inform* discourse action was carried out, but about *why* the action was performed, i.e. about the implication of the performed action.

applicability condition of *Obtain-Exemption* is that the agent not be a native North American, but CORE believes that EA *is* a native North American. Thus, CORE believes that *Obtain-Exemption(EA,French101)* is not feasible, resulting in the proposal being rejected.

4.1.2. Detecting suboptimal proposals. Joshi *et al.* (1984) and van Beek (1987) contended that a cooperative consultation system must go further than merely providing the user with the requested information. For instance, if there exists a better alternative to the proposal implicitly conveyed by the executing agent's question, instead of merely answering the question, the system should bring to the executing agent's attention the alternative plan for achieving his goal. However, since each agent has different characteristics and different beliefs about what constitutes a high-quality plan, a collaborative agent must take these factors into account when evaluating the validity and optimality of a proposal. One characteristic that strongly affects the evaluation of the optimality of a plan is the executing agent's preferences. Since in our collaborative planning environment the resultant plan is to achieve the executing agent's domain goal, will be carried out by the executing agent alone, and affects the executing agent but not the collaborating agent, the executing agent's preferences have a major impact on how good a particular plan is. Thus in the following discussion on optimality evaluation, we focus on evaluating the optimality of a proposal with respect to the executing agent's preferences.

Features of optimality evaluation: In order to tailor the system's evaluation of suboptimal plans to individual executing agents, we maintain, for each executing agent, a user model that includes, among other information, his *attribute-value preferences* (Elzer, Chu-Carroll & Carberry, 1994). An attribute-value preference indicates an agent's preferred value for a particular attribute of an object that might be used to instantiate a parameter of an action. For instance, with respect to the parameter *_course* in the action *Take-Course(_user,_course)*, an executing agent may prefer that the meeting time of *_course* be after 10 a.m. and that it be taught by Dr Smith. These preferences affect what the executing agent considers the *best* instantiation of *_course* and must be taken into account when the alternative instantiations are evaluated.

Clearly, agents consider some preferences to be more important than others; thus, in evaluating alternatives, an agent's preferences should not be treated equally. We associate with each preference a *strength* that indicates the importance the agent attaches to this particular preference. In evaluating alternatives, these preference strengths will enable a collaborative consultant to emphasize the preferences that the executing agent considers most important, and will hence allow the evaluation process to better address the executing agent's needs. For example, suppose a particular executing agent has a strong preference for courses taught by Dr Smith and a weak preference for courses that meet after 10 a.m. All other things being equal, this agent will presumably prefer taking CS360, which is taught by Dr Smith and meets at 9 a.m. to taking CS420, which is taught by Dr Brown and meets at noon. This is because CS360 satisfies the stronger preference while CS420 satisfies the weaker one.

However, it is not the case that an agent always considers a preference either satisfied or unsatisfied. Oftentimes, when none of the alternatives exactly satisfy an agent's preferences, she will prefer those that come *closer* to satisfying her preferences over others. Thus, in evaluating alternatives, it is important that a collaborative agent take into account the *closeness of the matches* between the attribute value preferred by the executing agent and the actual values of the attribute associated with the alternative being evaluated. For example, if the agent prefers taking easy courses, a moderate course should be considered preferable to a difficult one, even though neither of them exactly satisfies the agent's preference.

Based on the above analysis, we argue that in evaluating the optimality of a proposal, a collaborative agent must take into account both the strengths of the executing agent's preference as well as the closeness of the matches between the preferred and actual values of the attributes of the alternatives. The next section describes our *ranking advisor*, whose task is to determine the best instantiation of a variable given an executing agent's preferences. These preferences are represented in the form, prefers(_agent,_attribute (_object,_value),_action,_strength), which indicates that _agent has a _strength preference that the attribute _attribute of _object be _value when performing _action. For instance, *prefers(EA,Difficulty(_course,easy),Take-Course,strong(pos))* indicates that EA has a strong positive preference for taking easy courses. In this paper, we limit our discussion to the situation in which there are several possible instantiations of the parameters of an action (such as CS883 and CS889 instantiating the parameter _course in the action *Take-Course(_agent,_course)*) and do not consider instances in which there are several alternative generic actions that will accomplish a higher level action.

The ranking advisor: The ranking advisor's task is to determine how the parameters of an action can best be instantiated, based on the executing agent's preferences. For each object that can instantiate a parameter of an action (such as CS889 instantiating _course in *Take-Course(EA,_course)*), the ranking advisor is given the values of the object's attributes (e.g. *Difficulty(CS889,easy)*) and the executing agent's preferences with respect to possible values of these attributes (e.g. *prefers(EA, Difficulty(_course,easy), Take-Course, high-moderate(pos))*). The actual values of the attributes are obtained from the system's knowledge base; the executing agent's preferences are extracted from the system's model of the executing agent which is constructed incrementally by reasoning about the agent's utterances, the agent's acceptance/rejection of proposals, and stereotypical user preferences (Carberry, Chu-Carroll & Elzer, 1999).

In ranking candidate instantiations of a proposed action, the ranking advisor takes into account both the strengths of the executing agent's preferences, as well as the closeness of the matches between the preferred values and the actual values of the attributes. We utilize the *weighted additive rule* (Keeney & Raiffa, 1993; Payne, Bettman & Johnson, 1993) for modeling human decision-making in ranking the candidate instantiations. Given a set of attributes relevant to the agent's decision-making, the weighted additive rule assigns to each attribute a *weight* that indicates the importance of the attribute to the agent's overall decision, and to each candidate/attribute pair a *score* that represents how well the attribute's value for that particular candidate matches the agent's preferences. It then calculates an overall rating for a candidate by computing the product of the weight and the score for each attribute and summing the weighted scores for all attributes. The candidate with the highest overall rating will then be chosen. We adopt the weighted additive rule because it takes into account all of the relevant information about a candidate, resolves conflicting values by allowing tradeoffs to be made among attributes, and is often viewed as a normative procedure for evaluating alternatives based on preferential choices (Payne *et al*., 1993).†

In order to utilize the weighted-additive rule in our decision-making process, we must first define the *weights* and *scores* used to evaluate the alternatives. The strength of a preference indicates the importance of the preference to the executing agent and corresponds to the *weight* in the weighted additive rule. We model six degrees each of positive and negative preference strength (very-strong, strong, high-moderate, low-moderate, weak, and very-weak), where the identified strength of a preference is based on the surface form of an utterance used to convey a preference, the conversational circumstances in which the preference is conveyed, patterns of acceptance and rejection of proposals and a model of stereotypical user preferences (Carberry *et al*., 1999). The closeness of a match indicates how well the actual value of an attribute matches the value preferred by the executing agent; thus, it corresponds to the *score* assigned to a candidate/attribute pair in the weighted additive rule. The possible values for the closeness of a match are *exact*, *strong*, *weak* and *none*, because this is the minimum number of values needed for comparing the *closeness* of two attribute values when neither of them exactly matches the executing agent's preferred value. The closeness of a match is measured based on the *distance* between the actual and the preferred values of an attribute where the unit of measurement differs depending on the type of the attribute. An attribute may be such that a partial match is not possible (such as the professor teaching a course), in which case the closeness of the match is either *exact* or *none*. The values of some attributes may be arranged on a scale (such as the difficulty of a course), in which the closeness of a match is based on the distance between the two values compared with the distance between the end points of the scale. In our implementation, if the distance between the two values is greater than zero but less than or equal to 1/4 of the maximum distance, the two values have a *strong* match; if the distance is between 1/4 and 1/2 of the maximum distance, they have a *weak* match; and if the distance is greater than 1/2 of the maximum distance, the match between them is *none*. For example, for the difficulty of a course which is ranked on a scale of *very-difficult*, *difficult*, *moderate*, *easy* and *very-easy*, the distance between *very-difficult* and *moderate* is two, which is 1/2 of the maximum distance; thus, the match between the two values is *weak*. The values of other attributes may be arranged in a hierarchy (such as the content of a course), in which case the closeness of a match is measured based on the distance between the preferred value and the most specific common ancestor of the preferred and actual values, compared with the distance between the preferred value and the root node of the hierarchy (Chu-Carroll, 1996).

For each candidate instantiation, the ranking advisor assigns numerical values to the strength of the preferences for the relevant attributes and to the closeness of each match. Using the weighted additive rule, a weight is computed for each candidate instantiation by summing the products of corresponding terms of the strength of a preference and the closeness of a match. The weights are then normalized and the instantiation with the

† By utilizing the weighted-additive rule, which is a linear utility method, we assume that the executing agent's preferences are independent of one another. We believe this to be a reasonable assumption in the university course advisement domain. However, one can argue that an agent may prefer taking a course taught by Dr Brown only if his preference to take an AI course is satisfied (perhaps because Dr Brown is extremely good at teaching AI courses, but is quite unimpressive at other subjects). We leave it to future work to relax this assumption and investigate utilizing other more complex decision-making strategies.

Domain knowledge:

CS883:
Professor(CS883,Brown)
Meets-At(CS883,14:00–15:15)
Difficulty(CS883,moderate)
Workload(CS883,light)
Offered(CS883)

CS889:
Professor(CS889,Smith)
Meets-At(CS889,10:30–11:45)
Difficulty(CS889,easy)
Workload(CS889,moderate)
Offered(CS889)

User model information:
Prefers(EA,Professor(_course,White),Take-Course,strong(neg))
Prefers(EA,Meets-At(_course,before(12:00),Take-Course,low-moderate(pos))
Prefers(EA,Difficulty(_course,easy),Take-Course,high-moderate(pos))

FIGURE 3. Relevant domain knowledge and EA preferences.

TABLE 2
Evaluation of CS883 and CS889 based on EA's preferences

Attribute	Preference strength	CS883			CS889		
		Match		Weight	Match		Weight
Professor	Strong (neg) -5	None	0	0	None	0	0
Meets-At	Low-moderate (pos) 3	Weak	1	3	Exact	3	9
Difficulty	High-moderate (pos) 4	Strong	2	8	Exact	3	12
				11			21

highest normalized weight is then considered the *best* instantiation for the action under consideration.

Example of ranking two alternative instantiations: We demonstrate the ranking advisor by showing how two different instantiations, CS883 and CS889, of the *Take-Course* action are ranked. Figure 3 shows the system's knowledge about the attribute values for each of these two objects, as well as the system's beliefs about the executing agent's preferences with respect to the *Take-Course* action.

The ranking advisor matches the executing agent's preferences against the system's domain knowledge for each of CS883 and CS889. For each preference, the ranking advisor records the strength of the preference and the closeness of the match between the preferred value and the actual values for each of CS883 and CS889. For instance, in considering the attribute *difficulty*, the strength of the preference is *high-moderate(pos)*, and the closeness of the match is *strong* and *exact* for CS883 and CS889, respectively. Table 2 shows a summary of the preference strengths and the closeness of the matches for the relevant attributes for both candidates. The ranking advisor then assigns numerical values to each preference strength and closeness of match: values from 6 to -6 are assigned to preference strengths from *very-strong(pos)* to *very-strong(neg)*, and values from 3 to 0 are assigned to closeness of matches from *exact* to *none*. The preference strength and the closeness of match for each attribute are then multiplied to compute a weight for the attribute, and these attribute weights are summed to produce an overall evaluation for each alternative. A normalized evaluation for each alternative is then

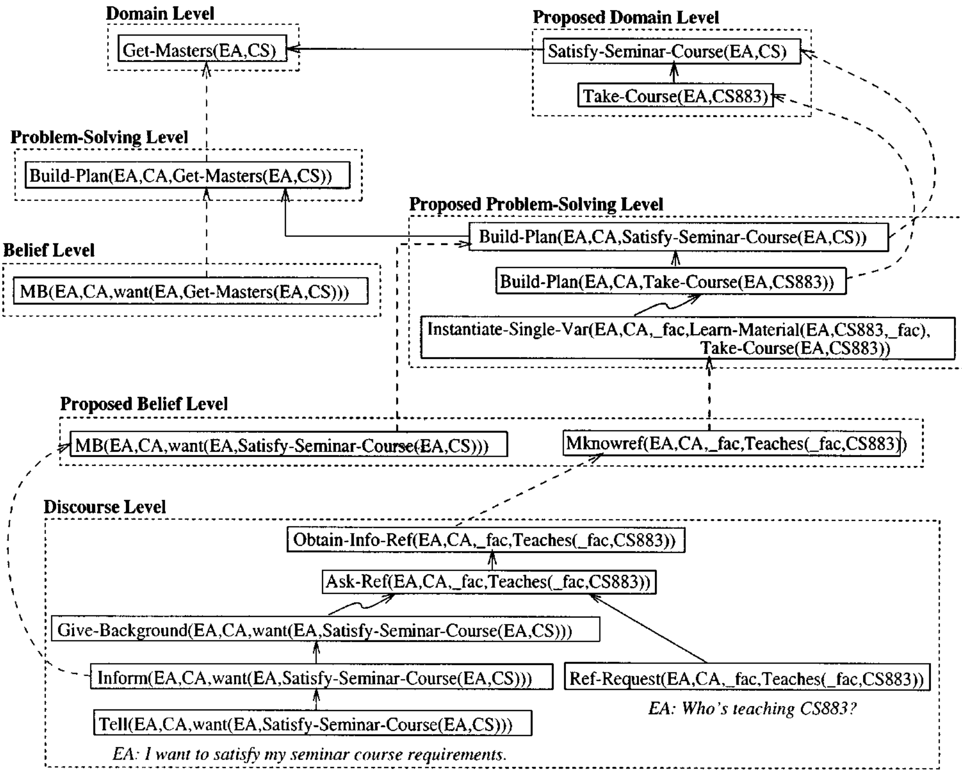


FIGURE 4. Dialog model for utterances (19) and (20).

calculated by dividing its overall evaluation by the maximum possible evaluation.† In this example, the normalized evaluation for CS883 is 11/21 and that for CS889 is 1, suggesting that CS889 is a better instantiation than CS883 for the *Take-Course* action for this particular executing agent.

Example of detecting suboptimal proposals: To illustrate how CORE makes use of agents' preferences in evaluating the optimality of a proposal, assume that EA has the preferences shown in Figure 3, that EA has conveyed his intention to obtain a Master's degree in Computer Science (*Get-Masters*(*EA*,*CS*)), and that EA then says the following:

(19) *EA*: *I want to satisfy my seminar course requirement.*
(20) *Who is teaching CS883*?

The dialog model for utterances (19) and (20), shown in Figure 4, indicates that EA is proposing the domain actions of taking CS883 as part of satisfying his seminar course requirement. Given the proposed actions and beliefs, CORE first evaluates the proposed

†The maximum possible evaluation is the evaluation that will be assigned to an alternative when the closeness of matches for all attributes with positive preferences is *exact*, and those for attributes with negative preferences are *none*.

domain actions starting at the top-level proposed action. CORE believes that *Satisfy-Seminar-Course(EA,CS)* is part of *Get-Masters(EA,CS)*, that *Satisfy-Seminar-Course(EA,CS)* is feasible, and that there is no alternative to *Satisfy-Seminar-Course(EA,CS)*; thus it evaluates its child action *Take-Course(EA,CS883)*. CORE believes that *Take-Course(EA,CS883)* contributes to *Satisfy-Seminar-Course(EA,CS)*, and that *Take-Course(EA,CS883)* is feasible; therefore, it evaluates the optimality of the proposed action. There are two instantiations of _course for *Take-Course(EA,_course)* that satisfy the constraints specified in the recipe for *Satisfy-Seminar-Course*: CS883 and CS889. The two alternatives are evaluated by the ranking advisor and the result of the evaluation, summarized in Table 2, suggests that CS889 is a substantially better alternative than CS883. Thus, EA's original proposal is not accepted due to a suboptimally instantiated parameter.

4.2. EVALUATING PROPOSED BELIEFS

Besides evaluating proposed domain and problem-solving actions, a collaborative agent must also determine whether or not beliefs proposed by other agents are consistent with her private beliefs. The belief level of our dialog model consists of one or more belief trees where the belief represented by a child node is intended to provide support for the belief represented by its parent.† The beliefs captured by the nodes may be of three forms: (1) *MB*(*_agent1*,*_agent2*,_prop), representing that _agent1 and _agent2 come to mutually believe _prop, (2) *Mknowref*(*_agent1*,*_agent2*,*_var*,*_prop*), meaning that _agent1 and _agent2 come to mutually know the referent of _var which will satisfy _prop, where _var is a variable in _prop and (3) *Mknowif*(*_agent1*,*_agent2*,*_prop*), representing _agent1 and _agent2 coming to mutually know whether or not _prop is true. *Inform* actions produce proposals for beliefs of the first type, while WH-questions and yes–no questions produce proposals for the second and third types of beliefs, respectively.‡

Previous research has noted that agents do not merely believe or disbelieve a proposition; instead, they often consider some beliefs to be stronger (less defeasible) than others (Lambert & Carberry, 1992; Cawsey, Galliers, Logan, Reece & Jones, 1993). Thus, we associate a *strength* with each belief by an agent; this strength indicates the agent's confidence in the belief being an accurate description of situations in the real world. The strength of a belief is modeled with *endorsements*, which are explicit records of factors that affect one's certainty in a hypothesis (Cohen, 1985), following (Cawsey *et al*., 1993; Logan *et al*., 1994). We adopt the endorsements proposed by Galliers (1992), based primarily on the source of the information, modified to include the strength of the informing agent's belief as conveyed by the surface form of the utterance used to express the belief. These endorsements are grouped into classes, *warranted*, *very-strong*, *strong*, *weak* and *very-weak*, based on the strength that each endorsement represents in order for

† In this paper, we only consider cases in which an agent's proposed pieces of evidence all uniformly support or attack a belief. In cases where an agent proposes evidence to attack a belief, the proposed belief tree will be represented as the pieces of evidence supporting the negation of the belief being attacked.

‡ Note that WH-questions *propose* that the agents come to mutually know the referent of a variable. Once the proposal is accepted, the agents will work toward achieving this. Mutual knowledge is established when the other agent responds to the question by providing the referent of the variable and the response is accepted by the questioner. Similarly for the case of yes–no questions.

the strengths of multiple pieces of evidence for a belief to combine and contribute to determining the overall strength of the belief.

As discussed earlier, beliefs proposed by an agent can be represented as a tree structure where beliefs represented by child nodes are intended to provide support for beliefs represented by their parent nodes. Given the belief tree(s) proposed by the executing agent, the system must determine whether or not to accept the belief(s) represented by the root node(s) of the tree(s) (henceforth referred to as the *top-level proposed belief(s)*). This is because the top-level proposed belief is the belief that contributes to the problem-solving actions, and thus affects the domain plan being constructed, while its descendents are only intended to provide support for establishing the top-level proposed belief. In evaluating a top-level proposed belief (_bel), the system must first gather all of its evidence for and against _bel. These pieces of evidence are obtained from three sources: (1) EA's proposal of _bel, (2) the system's own private evidence pertaining to _bel, and (3) evidence proposed by EA as support for _bel. However, the evidence proposed as support for _bel is only relevant to the system's evaluation of _bel if the proposed evidence is accepted by the system; thus, as part of evaluating _bel, the system must evaluate the pieces of evidence proposed as support for _bel, i.e. the descendents of _bel in the proposed belief tree, resulting in a recursive process. A piece of evidence for _bel consists of an antecedent belief and an evidential relationship between the antecedent belief and _bel. For example, one might support the claim that Dr Smith's research area is theory by stating that Dr Smith is teaching a theory seminar. This piece of evidence consists of the antecedent belief that Dr Smith is teaching a theory seminar and the evidential relationship that teaching a theory seminar generally implies that one's research area is theory.† To evaluate a piece of evidence, the system must evaluate both the antecedent belief and the evidential relationship. A piece of evidence is considered accepted if both the belief and the relationship are accepted, and rejected otherwise. Having evaluated each piece of evidence pertaining to _bel, the system then can combine the effects of these pieces of evidence and determine its belief about _bel.

Figure 5 presents our algorithm for evaluating a proposal of beliefs based on the above principles. **Evaluate-belief** is invoked with _bel instantiated as the top-level belief of a proposed belief tree. **Evaluate-Belief** calls a function **Determine-Acceptance** to compute whether to accept a belief. **Determine-Acceptance** uses a simplified version of Galliers belief revision mechanism‡ (Galliers, 1992; Logan *et al*., 1994) to determine whether or not a proposed belief or evidential relationship should be accepted or rejected based on the constructed evidence set. It makes this decision by separating the evidence into two

† In our model, an evidential relationship has two associated measures: (1) *degree*, representing the amount of support that the antecedent $\_bel_i$ provides for the consequent _bel, and (2) *strength*, representing how strongly the agent believes the evidential relationship (Chu-Carroll, 1996). For example, the system may have a very strong (strength) belief that a professor teaching a theory seminar provides very strong (degree) support for his research area being theory. Due to space limitations, we will not distinguish between *degree* and *strength* in the rest of this paper. We will use an agent's *strength of belief* in an evidential relationship to refer to *the amount of support that the agent believes the antecedent provides for the consequent*. This strength of belief is computed as the weaker of the *degree* and *strength* associated with the evidential relationship in the actual representation in our system.

‡ We adopt Galliers' belief revision mechanism for its ease of computation. The mechanisms that we have developed for evaluating proposed beliefs and for effectively resolving detected conflicts (Section 5.1) are independent of this belief revision mechanism. Readers are welcome to substitute their favorite means for combining beliefs of various strengths in its place.

Evaluate-Belief(_bel):

1. evidence set ← _bel (appropriately endorsed as conveyed by EA) and the system's private evidence that directly supports or attacks _bel
2. Evaluate each of _bel's children, _bel$_1$, … , _bel$_n$, (if any) in the proposed belief tree:
   2.1 /**evaluate antecedent belief _bel$_i$* */
   belief_result ← **Evaluate-Belief**(_bel$_i$)
   2.2. /**evaluate evidential relationship between _bel$_i$ and _bel* */
   relationship_result ← **Evaluate-Belief**(supports(_bel$_i$, _bel))
   2.3. If belief_result = relationship_result = accept, add {_bel$_i$,supports(_bel$_i$, _bel)} to the evidence set
   Else ignore _bel$_i$ and supports(_bel$_i$, _bel)
3. Return **Determine-Acceptance**(_bel, evidence set)

FIGURE 5. Algorithm for evaluating a proposed belief.

sets, one that contains evidence supporting _bel and one that contains evidence attacking _bel, and then comparing their respective strengths.† In determining the strength of a piece of evidence consisting of an antecedent belief and an evidential relationship, **Determine-Acceptance** follows Walker's *weakest link assumption* (Walker, 1992) and computes the strength of the piece of evidence as the weaker of the strengths of the antecedent belief and the evidential relationship.

Notice that the process for evaluating proposed beliefs (Figure 5) does not terminate as soon as a conflict in the agents' beliefs is detected. This is because the only proposed beliefs that are relevant to the domain plan being constructed are the top-level proposed beliefs. If these beliefs are agreed upon by both agents, it is irrelevant whether or not the agents agree on the evidence used to support them (Young, Moore & Pollack, 1994). Therefore, even if a child belief is rejected, the evaluation of its parent belief continues since acceptance of a parent belief is not contingent on the acceptance of the evidence proposed to support it.

4.2.1. Example of evaluating proposed beliefs. To illustrate the process for evaluating proposed beliefs, consider the following exchange:

(21) *CA*: *Dr Smith is going on sabbatical next semester.*
(22) *EA*: *Dr Smith is not going on sabbatical next semester.*
(23) *Dr Smith is teaching AI next semester.*

The dialog model for utterances (22) and (23), part of which is shown in Figure 6, suggests that EA is proposing three mutual beliefs: (1) Dr Smith is not going on sabbatical next semester, (2) Dr Smith is teaching AI next semester and (3) Dr Smith teaching AI provides support for the belief that he is not going on sabbatical.

†In this algorithm we make the assumption that one set of evidence is always stronger than the other; in other words, the system can always come to a decision as to whether to accept or reject _bel. However, our analysis of naturally occurring dialogs shows that in collaborative dialogs, agents are not always capable of making such a decision and in cases where an agent cannot determine whether to accept or reject a belief, she may initiate an *information-sharing subdialog* to share information with the other agent so that each agent can knowledgeably re-evaluate the belief. The process of initiating such information-sharing subdialogs during collaborative planning activities is discussed in Chu-Carroll and Carberry (1995*b*) and Chu-Carroll and Carberry (1998).

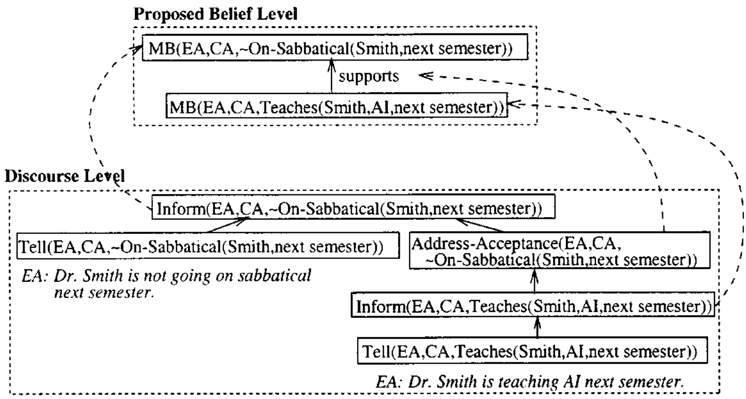


FIGURE 6. Belief and discourse levels for utterances (22) and (23).

The process of evaluating proposed beliefs is carried out by invoking the **Evaluate-Belief** algorithm on the top-level proposed belief, $\neg$ *On-Sabbatical*(*Smith*, *next semester*). Since evaluating the top-level proposed belief involves evaluating the evidence that EA proposed to support it (step 2 in Figure 5), CORE evaluates the only piece of evidence proposed by EA by invoking **Evaluate-Belief** on both the belief and the evidential relationship that constitute the evidence, namely *Teaches(Smith,AI,next semester)* and *supports(Teaches(Smith,AI,next semester),* $\neg$ On-Sabbatical*(Smith,next semester)).* When evaluating *Teaches(Smith,AI,next semester)*, CORE first searches in its private beliefs for evidence relevant to it. CORE very strongly believes that (1) Dr Brown is teaching AI next semester and (2) since only one person teaches a course, that Dr Brown teaching AI supports the belief that Dr Smith is not teaching AI. This pair of beliefs constitutes a very strong piece of evidence against *Teaches(Smith,AI,next semester)*, and is incorporated into the evidence set. In addition, EA's utterance in (23) conveys his belief that Dr Smith is teaching AI, which adds to the system's evidence set a strong piece of evidence that supports *Teaches(Smith,AI,next semester)*. This is only a strong piece of evidence because although EA stated the belief in a direct statement, he is not believed to be an expert in the area of course assignment; thus it is endorsed as {non-expert, direct-statement} which has a corresponding strength of strong. Since *Teaches*(*Smith,AI,next semester*) has no children in the belief tree, CORE uses just this evidence set when invoking the **Determine-Acceptance** function (step 3) to determine the acceptance of *Teaches(Smith,AI,next semester)*. Since the strength of the evidence against the proposed belief outweighs the strength of the evidence for the belief, the proposed belief is rejected.

Although CORE rejects the proposed belief *Teaches(Smith,AI,next semester)*, the evaluation process does not terminate but continues until the top-level proposed belief is evaluated. CORE next evaluates the proposed evidential relationship (step 2.2). Since CORE strongly believes that teaching a course implies that a faculty member is not on sabbatical, it accepts the proposed evidential relationship. Since CORE accepts *supports(Teaches(Smith,AI,next semester),* $\neg$ *On-Sabbatical(Smith,next semester))* but rejects *Teaches(Smith,AI,next semester)*, the evidence that EA proposed to support $\neg$ *On-Sabbatical(Smith,next semester)* is rejected by CORE (step 2.3). CORE then evaluates

¬ *On-Sabbatical(Smith,next semester)* itself (step 3). The evidence set contains a strong piece of evidence favoring Dr Smith not going on sabbatical, inferred from EA's statement in (22). The evidence set also contains a very strong piece of evidence that Dr Smith is going on sabbatical; this piece of evidence comes from CORE's private beliefs and is very strong because it was conveyed earlier via a direct statement by the chairperson of the department who is considered an expert in faculty members' whereabouts. Thus, the evidence in favor of the belief that Dr Smith is going on sabbatical outweighs the evidence against it, and CORE retains its original belief that Dr Smith is going on sabbatical next semester and rejects the top-level belief proposed by EA.

5. Collaborative negotiation for conflict resolution

The *collaborative planning principle* in Whittaker and Stenton (1988), Walker and Whittaker (1990) and Walker (1992) suggests that "conversants must provide evidence of a detected discrepancy in belief as soon as possible". Thus, once an agent detects a relevant conflict, she must notify the other agent of the conflict and attempt to resolve it—to do otherwise is to fail in her responsibilities as a collaborative participant. The *Modify* part of the *Propose–Evaluate–Modify* collaborative cycle is invoked if a conflict that warrants resolution is detected in the executing agent's proposal during the evaluation process. A conflict warrants resolution if it affects the domain plan being constructed; thus conflicts in the proposed domain and problem-solving actions and conflicts regarding the top-level proposed beliefs need to be resolved in order for the agents to effectively continue their collaborative planning. In our model, the modification process is carried out by invoking the *Modify-Proposal* problem-solving action, and its goal is for the agents to reach an agreement on accepting perhaps a variation of the executing agent's original proposal as a valid and reasonably efficient way of achieving the executing agent's domain goal. However, an agent would be considered uncooperative if she modifies a proposal without the other agent's consent; thus a collaborative agent must first convey to the other agent her desire to modify the proposal, and only when this proposal for modification is accepted will the modification process actually take place.

Communication for conflict resolution involves an agent (agent A) conveying to the other agent (agent B) the detected conflict and perhaps providing evidence to support her point of view. If agent B accepts A's proposed beliefs, thus accepting A's proposal to modify his original proposal, the actual modification of the proposal will be carried out. On the other hand, if B does not immediately accept A's claim, he might provide evidence to justify his point of view, leading to a negotiation subdialog to resolve the detected conflict. This negotiation subdialog may lead to (1) A accepting B's beliefs, thereby accepting B's original proposal and abandoning her proposal to modify it, (2) B accepting A's beliefs, allowing A to carry out the modification of the proposal or (3) a disagreement between A and B that cannot be resolved. The last case is beyond the scope of this paper.

A proposal for modification may fall into one of the three categories, based on features of the detected conflict. First, the conflict may be related to the validity of a proposition represented by a node in the proposal, i.e. when a proposed action is infeasible or when a proposed belief is rejected. Second, the conflict may be related to the validity of the relationship between two nodes in the proposal, i.e. when a proposed *contributes* relationship between two actions or a proposed *supports* relationship between two beliefs does not hold. Third, the conflict may be related to the optimality of the proposed plan, i.e. when a better alternative to the proposed plan exists. Notice that in cases where the conflict occurs at the domain or problem-solving level, the actual modification of the proposal involves either replacing a proposed action with an appropriate one, or replacing the instantiation of a parameter of a proposed action with an alternative instantiation. In some cases, however, either one of these two modifications will result in a valid plan. In such cases, we argue that the latter approach should be taken in order to retain as much of the executing agent's original proposal as possible. This is because the executing agent's proposal of a particular action indicates his desire to perform the action; therefore the action should not be completely replaced if it can be avoided.

Our model provides an operational semantics for an agent's *Modify* action by capturing these three categories of proposal modification in the recipe for the *Modify-Proposal* problem-solving action that is part of the recipe library used by CORE's mechanism for planning collaborative responses. A recipe (Pollack, 1986) is a template for performing an action. It includes a header specifying the action defined by the recipe, a type specification (decomposition, specialization or primitive), the applicability conditions, preconditions, and constraints of the action, the subactions comprising the body of the recipe and the goals of performing the action. The applicability conditions and preconditions are both conditions that must be satisfied before an action can be performed; however, it is anomalous for an agent to attempt to satisfy an unsatisfied applicability condition while it is reasonable for an agent to construct a plan to satisfy a failed precondition. Constraints limit the allowable instantiation of variables in each component of a recipe. The body of a recipe consists of a set of simpler subactions for performing the action encoded by the recipe. When the recipe type is *decomposition*, all subactions in the body of the recipe must be performed; when the recipe type is *specialization*, the actions in the body are alternative ways of performing the action given in the recipe's header; and when the recipe type is *primitive*, the body of the recipe is nil and the action can be performed immediately. Finally, the *goals* of an action are what the agent performing the action intends to achieve. Variables in recipes are represented as lowercase strings preceded by an underscore, with the string reflecting the variable's type; for example, _course1 and _course2 both refer to variables of type *course*.

The body of the recipe for the *Modify-Proposal* problem-solving action consists of three specializations: (1) *Correct-Node*, invoked when a proposed action is infeasible or when a proposed belief is not accepted; (2) *Correct-Relation*, invoked when a proposal is ill-formed or when the evidential relationship between two proposed beliefs is not accepted; and (3) *Improve-Parameter*, invoked when a better instantiation of a parameter is found. Each specialization eventually decomposes into some primitive action which actually modifies the proposal. However, since a collaborative agent should not modify a proposal without the other agent's consent, the three specializations share a common precondition—both agents must come to an agreement that the original proposal is faulty before any modification can take place. It is the attempt to satisfy this precondition that leads to the generation of natural language utterances to resolve the conflict in the agents' beliefs.

Figure 7 shows the recipes for *Correct-Node* and one of its subactions, *Modify-Node*. The applicability conditions of *Correct-Node* specify that the action can only be invoked when _agent1 believes that _prop, the proposition in the erroneous element _elem, is

Action:	**Correct-Node**(_agent1, _agent2, _elem, _proposed)
Type:	Decomposition
Appl Cond:	believe(_agent1, ¬acceptable(_prop))
	believe(_agent2, acceptable(_prop))
Const:	elem-type(_elem,node)
	prop-in(_prop,_elem)
Body:	Modify-Node(_agent1,_agent2,_proposed,_prop)
	Insert-Correction(_agent1,_agent2,_proposed)
Goal:	acceptable(_proposed)
Action:	**Modify-Node**(_agent1,_agent2,_proposed,_prop)
Type:	Specialization
Precond:	MB(_agent1,_agent2,¬acceptable(_prop))
Body:	Remove-Node(_agent1,_agent2,_proposed,_prop)
	Alter-Node(_agent1,_agent2,_proposed,_prop)
Goal:	modified(_proposed)

FIGURE 7. The *Correct-Node* and *Modify-Node* recipes.

not acceptable while _agent2 believes that it is acceptable, i.e. when the agents disagree about the feasibility of _prop when it is instantiated as an action or about the truth of _prop when it is instantiated as a belief. The precondition of *Modify-Node*, however, shows that the action can only be performed if _agent1 and _agent2 mutually believe that _prop is not acceptable — that is, the conflict between _agent1 and _agent2 must have been resolved. The attempt to satisfy this precondition causes the system to try to establish the mutual belief that _prop is not acceptable, leading it to invoke discourse actions to modify EA's beliefs. This can be viewed as the system initiating a negotiation subdialog to resolve the detected conflict. If the executing agent accepts the system's claim that the original proposal is erroneous, thus satisfying the precondition of *Modify-Node*, the original proposal can be modified; however, if the executing agent does not accept the system's beliefs, he may try to modify the system's (implicitly) proposed modification of his original proposal, resulting in a recursive modification process.

In order to retain as much of the original proposal as possible when modifying a proposal, the *Modify-Node* action has two specializations: *Remove-Node* and *Alter-Node*. *Remove-Node* is selected if the agents disagree about _prop in its entirety, i.e. if the generic action represented by _prop cannot be performed by the executing agent or if the belief represented by _prop is rejected by the system. It will cause the subtree rooted at the node that contains _prop to be removed from the dialog model. On the other hand, *Alter-Node* is selected if a parameter of the action represented by _prop is inappropriately instantiated. This will cause the generic action to remain in the dialog model but the problematic parameter to become uninstantiated. In subsequent dialog, the agents may propose replacements for the deleted portions of the dialog model by invoking *Insert-Correction*, the second subaction of *Correct-Node*.

5.1. SELECTING THE FOCUS OF MODIFICATION

When the *Modify-Proposal* action invokes one of its subactions, it must determine the aspect of the proposal that the system will address in its pursuit of conflict resolution. For instance, in the case of *Correct-Node*, it must determine how the parameter _elem in

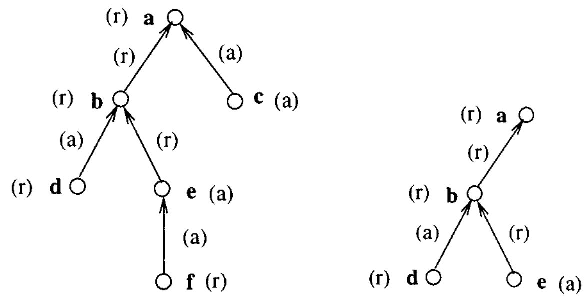
Correct-Node (Figure 7) should be instantiated. If the reason for proposal rejection occurs at the domain or problem-solving level, the focus of modification is the single action or *contributes* relationship identified as a source of disagreement during the evaluation process, since the evaluation of proposed actions terminates as soon as a conflict is detected. However, if the reason for rejection occurs at the belief level, determining the instantiation of this parameter is a more complicated process.

A proposal at the belief level is rejected if at least one of the top-level proposed beliefs is rejected. When a top-level proposed belief is rejected, the system may also have rejected some of the evidence proposed by EA to support the belief. For each rejected top-level proposed belief, _bel, the system could either attempt to change the executing agent's belief about _bel by (1) directly providing evidence against _bel itself, (2) changing the executing agent's beliefs about the rejected children to eliminate his reasons for believing in _bel and thereby causing him to accept $\neg$_bel or (3) addressing both _bel and its rejected children. Since collaborative agents are expected to engage in effective and efficient dialogs and not to argue for the sake of arguing, the system should address the rejected belief(s) that it predicts will most efficiently resolve the agent's conflict regarding the top-level proposed belief. This subset of rejected beliefs will be referred to as the *focus of modification.*

The process for selecting the focus of modification involves two steps. First, a collaborative agent must identify those beliefs and evidential relationships which, if refuted, might resolve the agents' conflict about the top-level proposed belief. Second, the agent must examine these beliefs and evidential relationships to select the subset that it will explicitly refute; this selection should be based on the likelihood of each potential choice changing the executing agent's belief about the rejected top-level proposed belief.

5.1.1. Identifying the candidate foci tree. We use the term *candidate foci tree* to refer to the set of beliefs whose refutation might resolve the agents' conflict about the top-level belief in a rejected proposed belief tree. The candidate foci tree is a tree structure that contains the rejected top-level proposed belief (since successful refutation of this belief will resolve the agents' conflict about the belief) as well as the pieces of evidence that satisfy the following two conditions. First, the evidence must have been rejected by the system, since if the agents agree about the evidence, then nothing would be gained by attempting to refute it. Second, the evidence must be intended to support a rejected belief or evidential relationship that is part of the candidate foci tree. This is because successful refutation of such evidence will lessen the support for the rejected belief or relationship and thus indirectly further refutation of the piece of evidence that it is part of; by transitivity, this refutation indirectly furthers refutation of the top-level proposed belief. For example, suppose that the system chooses to refute a piece of evidence E_2 that is intended to support a rejected belief in the candidate foci tree that is intended as evidence E_1 for the top-level belief. If this refutation convinces the other agent that E_2 is not valid evidence, then it reduces the amount of evidence supporting the belief in E_1 that was provided as support for the top-level belief; consequently, the conflict about this supporting belief in E_1 may be resolved, thereby providing less evidence for the top-level belief and potentially resolving the conflict about it.

Our algorithm for identifying the candidate foci tree captures these principles in a depth-first search of the nodes of a rejected proposed belief tree. Given the proposed



(a) (b)

FIGURE 8. An evaluated belief tree and its corresponding candidate foci tree.

belief tree shown in Figure 8(a), Figure 8(b) shows its corresponding candidate foci tree. The parenthesized letters indicate whether a belief or evidential relationship was accepted (a) or rejected (r) during the evaluation process. When traversing the evidence proposed to support node **a**, the piece of evidence containing **b** and **supports(b,a)** is incorporated into the candidate foci tree because the evidence itself was rejected by the system and the belief it is intended to support (node **a**) was also rejected. Similarly, **d** and **supports(d,b)** as well as **e** and **supports(e,b)** are included in the candidate foci tree. The evidence consisting of **c** and **supports(c,a)** is not included in the candidate foci tree because there is no conflict between the agents regarding the validity of this piece of evidence. Finally, **f** and **supports(f,e)** are not incorporated into the candidate foci tree because, even though the evidence itself was rejected by the system, both agents accept the belief that it is intended to support (node **e**); once the system accepts a belief, it is irrelevant whether it accepts the executing agent's evidence for that belief (Young *et al.*, 1994).

5.1.2. Selecting beliefs for explicit refutation. Once the candidate foci tree is identified, the system must determine the focus of modification for the top-level rejected belief—i.e. select the subset of the rejected beliefs in the candidate foci tree that will be explicitly refuted in order to resolve the agents' conflict about the top-level proposed belief. As discussed earlier, in order to resolve the agents' conflict about _bel, the system has three choices: (1) explicitly refute _bel by providing evidence against it, (2) explicitly refute the evidence that the executing agent proposed as support for _bel, thereby eliminating the executing agent's reasons for believing in _bel and (3) address both _bel and the evidence proposed to support it. Given these three choices, a collaborative agent's first preference should be to address the rejected evidence. The reason for this is twofold. First, McKeown's focusing rules suggest that continuing a newly introduced topic is preferable to returning to a previous topic (McKeown, 1985). When an agent refutes a piece of evidence proposed to support _bel, both the evidence and _bel are considered open beliefs and both beliefs can be addressed naturally in subsequent dialogs. On the other hand, if the agent addresses _bel directly by providing evidence refuting it, thus implicitly closing discussion of the pieces of evidence proposed to support _bel (even though some of them are not accepted by the agent), then it will be less coherent to return to these rejected pieces of evidence later on in the dialog. Second, in addressing a piece of rejected evidence to refute _bel, an agent conveys disagreement regarding both the evidence and _bel. If this refutation succeeds, then the agents not only have resolved their conflict about _bel, but have also eliminated a piece of invalid support for _bel. Although the agents' goal is only to resolve their conflict about _bel, removing support for _bel has the beneficial side effect of strengthening acceptance of it—i.e. removing any lingering doubts that the agent who gives up his belief in _bel might have. If the system chooses to refute the supporting evidence, then it must identify a minimally sufficient subset of this evidence to actually refute.

In order to determine whether or not the system can successfully refute _bel by addressing its rejected evidence, the system must first predict whether or not refuting the rejected evidence for _bel will result in the executing agent giving up his belief in _bel. This prediction process involves two steps. First, the system must predict whether or not it has sufficient justification to convince the executing agent that the pieces of evidence being refuted are invalid. If the system predicts that it can successfully refute a subset of the rejected evidence, the second step is to predict whether or not eliminating this subset of the rejected evidence is sufficient to cause the executing agent to accept ¬_bel. If refuting the rejected evidence is predicted to fail to resolve the agents' conflict about _bel, the system should consider whether directly attacking _bel will cause the executing agent to reject it. If this is again predicted to fail, the system should consider whether attacking both _bel and its children will cause the executing agent to reject _bel. If none of these is predicted to succeed, then the system does not have sufficient evidence to convince the executing agent of ¬_bel.

Our algorithm **Select-Focus-Modification** shown in Figure 9 captures the aforementioned principles in analysing the candidate foci tree and selecting the beliefs that will be explicitly refuted. This selection process requires that the system be able to predict the effect that a set of evidence will have on the executing agent's acceptance of a belief. Logan *et al.* (1994) proposed a mechanism for predicting how a hearer's beliefs will be altered by some communicated beliefs. This mechanism utilizes Galliers' belief revision mechanism (Galliers, 1992) to predict the hearer's belief in _bel based on two sources of information: (1) the speaker's beliefs about the hearer's evidence pertaining to _bel and (2) the evidence that the speaker is planning on presenting to the hearer. Information from these two sources captures the speaker's beliefs about the hearer's evidence for and against _bel (after the speaker's evidence relevant to _bel has been presented to the hearer); by applying Galliers' belief revision mechanism to these pieces of evidence, the speaker is then able to predict the hearer's resulting belief about _bel. Our **Select-Focus-Modification** algorithm invokes a function **Predict** that utilizes this strategy for predicting the hearer's beliefs.

If our algorithm determines that resolving the conflict about _bel involves refuting its rejected evidence (steps 4.2 and 4.4), then it invokes the **Select-Min-Set** function to identify a minimally sufficient subset of the rejected beliefs that will actually be refuted. **Select-Min-Set** first takes those pieces of evidence in _cand-set and orders them in decreasing order of the impact that each piece of evidence is believed to have on the strength of the executing agent's belief in _bel. In order to select the minimum subset that is sufficient for changing the executing agent's belief about _bel, the system first predicts whether or not changing the executing agent's belief about the first piece of evidence

Select-Focus-Modification(_bel):

1. _bel.u-evid ← system's beliefs about EA's evidence pertaining to _bel
   _bel.s-attack ← system's own evidence against _bel

2. If _bel is a leaf node in the candidate foci tree,
   2.1. If **Predict**(_bel, _bel.u-evid + _bel.s-attack)) = reject
        then _bel.focus ← {_bel}; return
   2.2 Else _bel.focus ← nil; return

3. /**Select focus for each of _bel's children in the candidate foci tree, _bel$_1$, … , _bel$_n$:* */
   3.1. If supports(_bel$_i$, _bel) is accepted but _bel$_i$ is not, **Select-Focus-Modification**(_bel$_i$).
   3.2. Else if _bel$_i$ is accepted but supports(_bel$_i$, _bel) is not, **Select-Focus-Modification-**(supports(_bel$_i$, _bel)).
   3.3. Else **Select-Focus-Modification**(_bel$_i$) and **Select-Focus-Modification**(supports(_bel$_i$, _bel))

4. /**Choose between attacking the proposed evidence for _bel and attacking _bel itself:* */
   4.1. /* *Form candidate set consisting of the pieces of evidence that the system rejected and which it predicts it can successfully refute* */
        _cand-set ← {{_bel$_i$,supports(_bel$_i$, _bel)} | rejected({_bel$_i$,supports(_bel$_i$, _bel)}) ∧
        (¬rejected(_bel$_i$) ∨ _bel$_i$.focus ≠ nil) ∧
        (¬rejected(supports(_bel$_i$, _bel)) ∨
        supports(_bel$_i$, _bel).
        focus ≠ nil)}
   4.2. /**Check if addressing _bel's rejected evidence is sufficient* */
        If **Predict**(_bel, _bel.u-evid − _cand-set) = reject (i.e., EA's disbelief in all rejected evidence which the system can refute will cause him to reject _bel),
        min-set ← **Select-Min-Set**(_bel, _cand-set)
        _bel.focus ← $\bigcup_{\_bel_i \in \_min\text{-}set}$ _bel$_i$.focus
   4.3. /* *Check if addressing _bel itself is sufficient* */
        Else if **Predict**(_bel, _bel.u.evid + _bel.s-attack)) = reject (i.e., the system's evidence against _bel will cause EA to reject _bel),
        _bel.focus ← {_bel}
   4.4. /* *Check if addressing both _bel and its rejected evidence is sufficient* */
        Else if **Predict**(_bel, _bel.s-attack + _bel.u-evid − _cand-set)) = reject
        min-set ← **Select-Min-Set**(_bel, _cand-set∪{_bel})
        _bel.focus ← $\bigcup_{\_bel_i \in \_min\text{-}set\text{-}\{\_bel\}}$ _bel$_i$.focus∪{_bel}
   4.5. Else _bel.focus ← nil

FIGURE 9. Algorithm for selecting the focus of modification.

(_evid$_1$) is sufficient. If not, this indicates that merely addressing one piece of evidence will not be sufficient to change the executing agent's belief about _bel (since the other pieces of evidence contribute less to the executing agent's belief in _bel); thus the system predicts whether addressing the first two pieces of evidence in the ordered set is sufficient. This process continues until the system finds the first n pieces of evidence which it predicts, when disbelieved by the executing agent, will cause him to give up his belief in _bel. The components of these n pieces of evidence that were rejected by the system are then returned by **Select-Min-Set**, and the focus of modification for _bel is the union of the focus for each element in _min-set (step 4.2). This process guarantees that _min-set is the minimum subset of evidence proposed to support _bel that the system believes it must address in order to change the executing agent's belief in _bel.

After the **Select-Focus-Modification** process is completed, each rejected top-level proposed belief (_bel) will be annotated with a set of beliefs on which the system should focus (_bel.focus) when attempting to change the executing agent's view of _bel. The negations of these beliefs are then posted by the system as mutual beliefs to be achieved in order to carry out the modification process, leading the system to generate natural language utterances to convey these beliefs and to provide supporting evidence for them.†

5.2. EXAMPLES OF COLLABORATIVE NEGOTIATION FOR CONFLICT RESOLUTION

5.2.1. Correcting invalid proposals. In order to illustrate the process for modifying proposed actions, we return to the example in utterances (17) and (18), repeated below, whose dialog model is shown in Figure 2.

(17) *EA*: *I want to satisfy my foreign language requirement.*
(18) *Where is the exemption form for French101?*

The evaluation of the proposed domain actions in Figure 2 was discussed in Section 4.1.1, and resulted in the rejection of the proposal because CORE believes that the proposed action *Obtain-Exemption*(*EA*,*French*101) is not feasible.

Since EA's proposal is not accepted, CORE invokes the *Modify-Proposal* action in order to resolve the detected conflict in the proposal. Since the only detected conflict is the feasibility of *Obtain-Exemption(EA,French101)*, it is selected as the focus of modification, thus leading *Modify-Proposal* to select *Correct-Node* (Figure 7) as its specialization. Figure 10 shows the modification process and how the *Correct-Node* action is expanded. In order to satisfy the precondition of *Modify-Node* that both agents agree about the infeasibility of *Obtain-Exemption(EA,French101)*, CORE attempts to establish the mutual belief that *Obtain-Exemption(EA,French101)* is infeasible and to support it by establishing the mutual belief that EA is a native North American. To accomplish this, CORE invokes *Inform* discourse actions (Figure 11), whose goal is for the agents to mutually believe a proposition, thus generating the semantic representations of the following two utterances:

(24) *CA*: *Obtaining an exemption for French101 is not feasible.*
(25) *You are a native North American.*

Notice that in Figure 10, the actions for modifying EA's proposal operate on the entire dialog model in Figure 2, and therefore are represented as meta-level problem-solving actions. If EA accepts CORE's proposed beliefs, thus satisfying the precondition of *Modify-Node*, the modification process can be performed and changes made to EA's original proposal. In this example, EA's proposal was rejected because the generic action *Obtain-Exemption* could not be performed by EA; thus, once EA has agreed that the proposal must be modified, *Remove-Node* will be selected as a specialization of *Modify-Node*, and *Obtain-Exemption* and all proposed actions and mutual beliefs that contribute to it will be removed from the proposal. Alternative actions that contribute to *Satisfy-Foreign-Language(EA)* (such as *Take-Course(EA,French101)*) may be incorporated into the proposal by *Insert-Correction*, the second subaction of *Modify-Node*.

Assuming that the agents encounter no further conflict in performing *Insert-Correction*, the modification process at the meta-level is completed and the modified base-level dialog model is returned to. The proposed additions now consist of actions agreed upon

† For details on how the system selects a subset of the available pieces of evidence to present to the executing agent as support for the system's claims, see Chu-Carroll and Carberry (1998) and Chu-Carroll (1996).

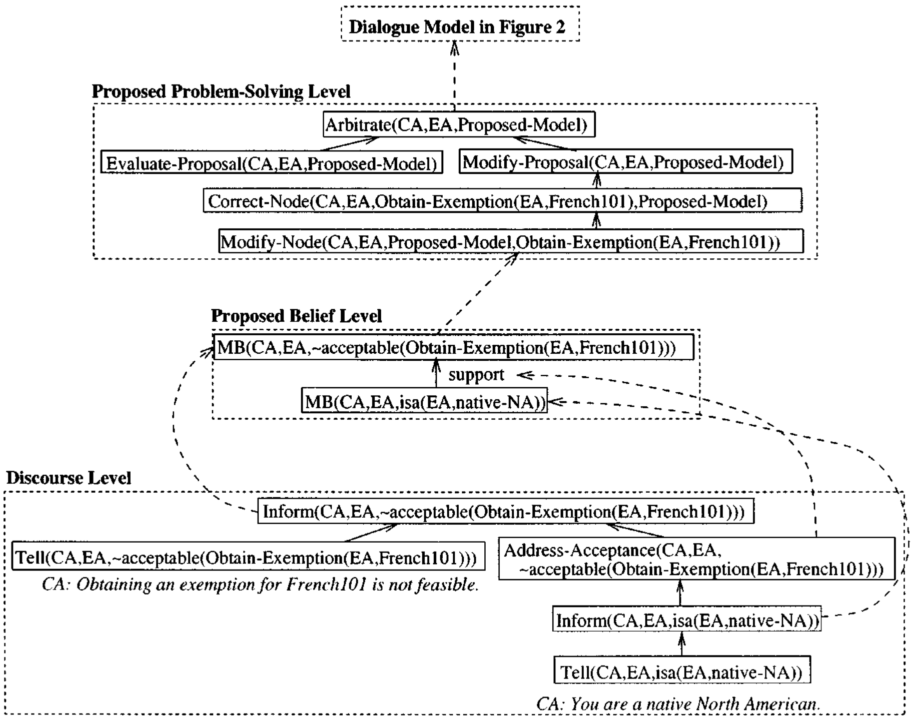


FIGURE 10. Responding to implicitly conveyed conflicts.

Action:	**Inform**(_agent1, _agent2, _prop)
Type:	Decomposition
Appl Cond:	believe(_agent1, _prop) believe(_agent1, ¬ believe(_agent2, _prop))
Body:	Tell(_agent1, _agent2, _prop) Address-Acceptance(_agent1, _agent2, _prop)
Goal:	MB(_agent1, _agent2, _prop)

FIGURE 11. Recipe for the *Inform* discourse action.

by both agents and will therefore be incorporated into the existing model. Notice that our model separates the negotiation subdialog (captured at the meta level) from the original dialog while allowing the same plan-based mechanism to be used at both levels. Furthermore, since *Remove-Node* removes all proposed actions and beliefs that contribute to *Obtain-Exemption(EA,French*101*)*, including the proposal at the belief level that the agents mutually know the referent of the location where there are exemption forms for *French101*, EA's question in utterance (18) will no longer be answered. Thus, our model accounts for a feature of collaborative planning dialogs—questions may never be answered because they become superfluous to a correct means of achieving an agent's goals.

5.2.2. Suggesting better alternatives. To illustrate how CORE is capable of suggesting better alternatives to agents' proposals, we return to the example shown in utterances (19) and (20), repeated below:

(19) *EA*: *I want to satisfy my seminar course requirement.*
(20) *Who is teaching CS883*?

The evaluation of the dialog model for these utterances (Figure 4) was discussed in Section 4.1.2, and led to the rejection of EA's proposal because CORE believes that there exists a better alternative.

CORE then invokes the *Modify-Proposal* action which selects as its specialization *Improve-Parameter*. Similar to the previous example, CORE attempts to satisfy the precondition that the agents' beliefs be reconciled before actually modifying the proposal; thus it posts the belief that CS889 is a better alternative than CS883 as a mutual belief to be achieved, supported by its reasons for holding this belief. The supporting evidence is selected by comparing the sets of information used by the ranking advisor (Table 2) and selecting the features that contribute the most to making CS889 preferable to CS883. This process leads the system to generate the semantic representations of the following utterances:

(26) *CA*: *CS889 is a better alternative than CS883.*
(27) *The difficulty level of CS889 is easy.*
(28) *CS889 meets at 10:30 am.*

5.2.3. Addressing conflicting beliefs. To illustrate the process for resolving conflicts in proposed beliefs, consider the case in which EA responds to CA's utterances in (24) and (25) as follows.

(29) *EA*: *I am not a native North American.*
(30) *I was born in Paris.*

Utterances (29) and (30) would be interpreted as an indication that EA does not accept CA's previously proposed mutual belief *isa(EA,native-NA)*. Figure 12 shows the dialog model that would be constructed after these utterances.

Suppose CORE believes that EA has been living in North America since age five and that for the purpose of foreign language exemptions, a person is considered a non-native North American only if he (1) was born outside of North America and (2) lived outside of North America until at least age six. CORE evaluates the proposed beliefs by invoking **Evaluate-Belief** on the top-level proposed belief $\neg$*isa(EA,native-NA)*. **Evaluate-Belief** first evaluates the evidence proposed to support $\neg$*isa(EA,native-NA)*, namely *born(EA,Paris) and supports(born(EA,Paris),* $\neg$*isa(EA,native-NA))*. CORE accepts the belief that EA was born in Paris, and evaluates the proposed evidential relationship. CORE believes that in order for one to be considered a non-native North American, conditions (1) and (2) above must hold. Although EA explicitly stated condition (1) (in utterance (30)), CORE believes that the second condition necessary for the evidential relationship to hold is not true (since CORE believes that EA has been living in North America since age five); thus, it rejects the proposed evidential relationship. Consequently, CORE rejects EA's proposed evidence for $\neg$*isa(EA,native-NA)*, thus retaining

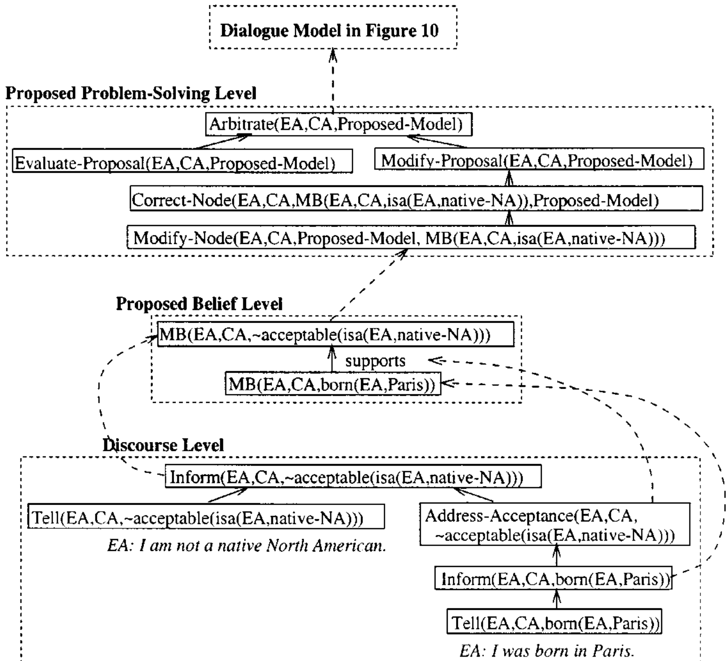


FIGURE 12. Dialog model for utterances (29) and (30).

its prior belief that EA *is* a native North American (as conveyed in utterance (25)). Since EA's top-level proposed belief is rejected, the modification process will be invoked.

In modifying the proposal, the focus of modification must be identified in order to determine how the subactions of *Modify-Proposal* should be selected and instantiated. In selecting the focus of modification, CORE first identifies the candidate foci tree and then invokes the **Select-Focus-Modification** algorithm on the belief at the root node of the candidate foci tree. In this example, the candidate foci tree is identical to the proposed belief tree in Figure 12, since both the top-level proposed belief and the evidence proposed to support it were rejected during the evaluation process. This indicates that the focus of modification could be either $\neg$ *isa(EA,native-NA)* or the *supports* relationship between the two proposed beliefs (since *born(EA,Paris)* was accepted by CORE). When **Select-Focus-Modification** is applied to $\neg$ *isa*(*EA,native-NA*), the algorithm is first recursively invoked on *supports(born(EA,Paris),* $\neg$ *isa(EA,native-NA))* to determine the focus for modifying the evidential relationship (step 4.2 in Figure 9). CORE has a warranted† belief that in order to be considered a non-native North American, one must have been born outside of North America and have lived outside of North America until at least age six. Since CORE is an authority in the area of university policy, it predicts that these beliefs, when presented to EA, will be sufficient to change his belief in the proposed evidential relationship; thus the focus of modification for this evidential

† *Warranted* is the strongest level of belief; in our system, CORE's beliefs about university policies and procedures are encoded as warranted beliefs, since the system has authoritative knowledge in this area.

Action:	**Correct-Relation**(_agent1,_agent2_elem,_proposed)
Type:	Decomposition
Appl Cond:	believe(_agent1, ¬ holds(_rel,_prop1,_prop2))
	believe(_agent2, holds(_rel,prop1,_prop2))
Const:	elem-type(_elem,link)
	relation-name(_elem,_rel)
	parent-node(_elem,_prop2)
	child-node(_elem,_prop1)
Body:	Modify-Relation(_agent,_agent2,_proposed,_rel,_prop1,_prop2)
	Insert-Correction(_agent1,_agent2,_proposed)
Goal:	well-formed(_proposed)
Action:	**Modify-Relation**(_agent1,_agent2,_proposed,_rel,_prop1,_prop2)
Type:	Specialization
Precond:	MB(_agent1,_agent2,¬ holds(_rel,_prop1,_prop2))
Body:	Remove-Node(_agent1,_agent2,_proposed,_prop1)
	Alter-Node(_agent1,_agent2,_proposed,_prop1)
Goal:	modified(_proposed)

FIGURE 13. The *Correct-Relation* and *Modify-Relation recipes.*

relationship is the relationship itself (step 2). Having selected the focus of modification for the rejected evidential relationship, CORE then determines whether or not attacking this evidential relationship is sufficient to change EA's belief about the top-level proposed belief (step 4.2). Since CORE believes that having been born in Paris is the only piece of evidence EA has for his belief that he is not a native North American, it predicts that eliminating EA's belief in *supports(born(EA,Paris), ¬ isa(EA,native-NA))* will cause him to give up on his belief in ¬ *isa(EA,native-NA)*; thus, the focus of modification will be the rejected evidential relationship.

Since the focus of modification is an evidential relationship, *Correct-Relation* is selected as a specialization of the *Modify-Proposal* action. Figure 13 shows the recipes for *Correct-Relation* and one of its subactions, *Modify-Relation*. Notice that the applicability conditions of *Correct-Relation* indicate that the action can be invoked if _agent1 believes that the relationship between _prop1 and _prop2 does not hold (in this example the system believes that the supports relationship between *born(EA,Paris)* and ¬ *isa(EA,native-NA)* does not hold) while _agent2 believes that this relationship does hold. The precondition of *Modify-Relation*, however, shows that the action can be performed only if _agent1 and _agent2 mutually believe that the relationship in question is invalid, i.e. the conflict between the agents must have been resolved. Thus, CORE posts ¬ *holds(supports,born(EA,Paris),* ¬ *isa(EA,native-NA))* as a mutual belief to be achieved between the agents. Discourse actions are then invoked to convey the belief as well as its supporting evidence to EA, leading to the generation of the semantic representations of the following utterances:

(31) *CA*: *Having been born in Paris does not support you being considered a non-native North American.*

(32) *To be considered a non-native North American for the purpose of foreign language exemption, you must both have been born outside of North America and have lived outside of North America until at least age six.*

Now if EA conveys acceptance of the proposed mutual beliefs, thereby satisfying the precondition of *Correct-Relation*, the proposed evidential relationship at the belief level in Figure 12 will be removed. This leaves EA with no more reason to believe that he is not a native North American;† thus, the *Modify-Proposal* action and its subactions in Figure 12 will be abandoned and the dialog model in Figure 10 will be returned to. EA accepts the proposed mutual belief that he is a native North American and thus the mutual belief that he is not eligible for a foreign language exemption, the latter of which satisfies the precondition of *Modify-Node*. Therefore, CORE executes *Modify-Node* and removes the *Obtain-Exemption* action and all actions and beliefs that contribute to the proposal of *Obtain-Exemption* in Figure 2.

6. Implementation and Evaluation

6.1. SYSTEM IMPLEMENTATION

We have implemented a prototype of our conflict resolution system, CORE, for a university course advisement domain; the system was implemented in Common Lisp with the Common Lisp Object System under SunOS. CORE realizes the response generation process for conflict resolution by utilizing the strategies detailed in this paper. Given the dialog model constructed from EA's proposal, it performs the evaluation and modification processes in our *Propose–Evaluate–Modify* framework. Domain knowledge used by CORE includes (1) knowledge about objects in the domain, their attributes and corresponding values, such as the professor of CS882 being Dr Smith, (2) knowledge about a hierarchy of concepts in the domain; for instance, *computer science* can be divided into *hardware*, *software* and *theory* and (3) knowledge about evidential inference rules in the domain, such as *a professor being on sabbatical normally implies that he is not teaching courses*. CORE also maintains a user model for each executing agent. The user model contains three types of information: (1) EA's preferences associated with actions in the domain, (2) EA's particular circumstances and characteristics relevant to the domain, such as EA having obtained credit for CS601 and (3) CORE's beliefs about EA's domain knowledge. Both CORE's domain knowledge and its beliefs about EA as captured in its user model will be used in its evaluation of a proposal made by EA, and in tailoring its responses to the particular EA. In addition, CORE maintains a library of generic recipes in order to plan its actions. In our implementation, CORE has knowledge about 29 distinct objects, 14 evidential rules, and 43 domain, problem-solving and discourse recipes. Since the focus of this work is on the evaluation and modification processes which are captured as problem-solving actions, 25 of the 43 recipes are domain-independent problem-solving recipes.

CORE takes as input a four-level dialog model that represents intentions inferred from EA's utterances, such as that in Figure 2. This dialog model is hand-generated based on what would be the output of the hypothesized plan recognition algorithm discussed in Section 3. It then evaluates the proposal to determine whether to accept the proposal or to reject the proposal and attempt to modify it. As part of the conflict resolution process,

† If EA had other reasons to believe that he is not a native North American, he, as a collaborative agent, would provide them as further support for this belief, instead of merely accepting utterances (31) and (32).

CORE determines the discourse acts that should be adopted to respond to EA's utterances, and generates the semantic forms of the utterances that realize these discourse acts.

6.2. EVALUATION OF CORE

6.2.1. Methodology. In order to obtain an initial assessment of the quality of CORE's responses, we performed an evaluation to determine whether or not the strategies adopted by CORE are reasonable strategies that a system should employ when participating in collaborative planning dialogs and whether other options should be considered. The evaluation, however, was not intended to address the completeness of the types of responses generated by CORE, nor was it intended to be a full-scale evaluation such as would be provided by integrating CORE's strategies into an actual interactive advisement system.

The evaluation was conducted via a questionnaire in which human judges ranked CORE's responses to EA's utterances among a set of alternative responses, and rated their level of satisfaction with each individual response. The questionnaire contained a total of seven dialog segments that demonstrated CORE's ability to correct invalid plans, to suggest better alternatives, and to correct conflicting beliefs; other dialog segments included in the questionnaire addressed aspects of CORE's performance that are not relevant to the topic of this paper. For each dialog segment, the judges were given the following information:

- *Input to CORE*: This included EA's utterances (for illustrative purposes), the actions and beliefs that would be inferred from each of these utterances and the relationship among them. In effect, this is a textual description of (a relevant subset of) the dialog model that would be inferred from EA's utterances.
- *CORE's relevant knowledge*: CORE's knowledge relevant to its evaluation of the actions/beliefs given in the input, CORE's strength of belief in each piece of knowledge, as well as CORE's beliefs about EA's preferences and their strengths.
- *Responses*: For each dialog segment, five alternative responses were given, one of which was the actual response generated by CORE (the responses were presented in random order so that the judges were not aware of which response was actually generated by the system). The other four responses were obtained by altering CORE's response generation strategies. For instance, the response generation strategies for correcting invalid plans may be modified to provide answers to EA's questions before correcting the proposal. Similarly, when conflict exists about a top-level belief in a proposed belief tree, the preference in the **Select-Focus-Modification** algorithm can be altered to allow CORE to consider directly refuting the top-level belief before considering refuting its rejected evidence. Appendix A shows a sample dialog from the questionnaire to illustrate the degree of variation of the alternative responses and how CORE's generation algorithm was modified to obtain these responses.

In the questionnaire, the judges were explicitly instructed not to pay attention to the phrasing of CORE's responses, but to evaluate the responses based on their *conciseness*, *coherence* and *effectiveness*, since it was the *content* of CORE's responses that was of interest in this evaluation. Based on this principle, the judges were asked to rate the five responses in the following two ways:

1. *Level of Satisfaction*: The goal of this part of the evaluation was to access the level of satisfaction that a user interacting with CORE is likely to have based on CORE's responses. Each alternative response was rated on a scale of *very good*, *good*, *fair*, *poor* and *terrible*.
2. *Ranking*: The goal of this ranking was to compare our response generation strategies with other alternative strategies that might be adopted in designing a response generation system. The judges were asked to rank in numerical order the five responses based on their evaluation of the quality of the responses.

Twelve subjects, all of whom are undergraduate or graduate students in computer science or linguistics, were asked to participate in this evaluation; evaluation forms were returned anonymously by 10 subjects by the established deadline. Note that the judges had not been taught about the CORE system and its processing mechanisms prior to the evaluation.

6.2.2. Results. Two sets of results were computed for the subjects' level of satisfaction with CORE's responses, and for the ranking of CORE's responses as compared with the alternative responses. The results of our evaluation are shown in Tables 3(a) and 3(b). In

TABLE 3
Evaluation results

(a) *Satisfaction rating*

	Mean of CORE's responses	Std. dev. of CORE's responses	Mean of all other responses
IP1	3.8	0.87	2.8
IP2	3.1	0.83	3.1
IP3	3.9	0.7	2.3
BA1	3.1	0.94	3.25
BA2	3.0	1.0	3.18
CB1	3.0	0.63	2.85
CB2	3.8	0.60	2.65

(b) *Ranking*

	Mean of CORE's responses	Std. dev. of CORE's responses	Ranking of CORE's mean
IP1	1.9	0.70	2
IP2	3.1	0.70	4
IP3	1.5	0.67	1
BA1	3.1	0.54	3
BA2	3.0	0.45	3
CB1	2.9	0.83	3
CB2	1.8	0.40	2

order to assess the judges' level of satisfaction with CORE's responses, we assigned a value of 1–5 to each of the satisfaction ratings where 1 is *terrible* and 5 is *very good*. The mean and standard deviation of CORE's actual response in each dialog segment were then computed, as well as the mean of all alternative responses provided for each dialog segment, which was used as a basis for comparison. Table 3(a) shows that in the three dialog segments in which CORE corrected an invalid plan (IP1, IP2 and IP3), the mean satisfaction ratings given to CORE's actual response range from being as good as to being substantially better than the mean satisfaction ratings given to all other responses [columns 1 and 3 in Table 3(a)]. However, the two dialog segments in which CORE suggested a better alternative (BA1 and BA2) were shown to be more problematic in that the means of CORE's responses were lower than the mean of all alternative responses and that the judges' ratings were less uniform than in the other dialogs (as shown by the standard deviations). Finally, the judges found CORE's responses in the two dialog segments in which CORE corrected detected conflicts in beliefs (CB1 and CB2) to be more coherent, concise and effective than the alternative responses.

To assess the ranking of CORE's responses as compared with alternative responses, we again computed the means and standard deviations of the rankings given to CORE's responses, as well as the mean ranking given to each of the alternative responses. The first column in Table 3(b) shows the mean rankings of CORE's responses. This set of results is consistent with that in Table 3(a) in that the dialog segments in which CORE's responses received a higher mean satisfaction rating also received a lower mean ranking (thus indicating a higher preference). The last column in Table 3(b) shows how the mean of CORE's response in a dialog segment ranks when compared to the mean rankings of its alternative responses in the same dialog segment. A comparison between columns 1 and 3 in Table 3(b) shows that, in each dialog segment, the mean ranking of CORE's response corresponds approximately to the ranking of CORE's mean among the mean rankings of all responses. The only exception is dialog IP2 where the mean of CORE's response is 3.1 while the ranking of CORE's mean is 4. For this dialog segment, the mean rankings for the four alternative responses are 3.0, 4.9, 1 and 3.0, respectively, indicating that the judges were consistent in selecting the best and worst responses, but did not have a clear preference between the other three responses, one of which was CORE's actual response.

Next, we attempted to examine the alternative responses that are consistently ranked higher than CORE's responses in each dialog segment. However, since many responses in the same dialog segment received similar mean satisfaction ratings, we performed the χ^2 test between the satisfaction ratings given to CORE's actual response and each alternative response. We then selected for our analysis only those responses (1) whose satisfaction rating scores are significantly different from CORE's ratings ($p < 0.05$) and (2) which are also ranked higher in preference than CORE's response. Our analysis of these selected alternative responses is summarized in Table 4. For the dialog segments in which CORE corrected invalid plans or suggested better alternatives, we compared the preferred response to CORE's actual response based on their agreement on the outcome of the proposal evaluation, whether or not the user's question is directly answered and whether or not additional (unsolicited) information is provided in the responses, as shown in Table 4(a). The table shows that in the three dialog segments where the top-ranked response received significantly better scores than CORE's response, the preferred response is produced as a result of the evaluation component having

TABLE 4
Comparison of CORE's responses with preferred responses

(a) *Dialog segments involving rejected actions*

	Proposal evaluation		Question answered?		Additional information?	
	Preferred response	CORE	Preferred response	CORE	Preferred response	CORE
IP2	reject	reject	yes	no	yes	yes
BA1	reject	reject	yes	no	yes	yes
BA2	reject	reject	yes	no	yes	yes

(b) *Dialog segments involving rejected beliefs*

	Evaluate-Belief		Select-Focus-Modification	
	Preferred response	CORE	Preferred response	CORE
CB1	reject	reject	all	child

rejected the proposal (which is in agreement with CORE), of the response generation component having directly answered the user's question (as opposed to CORE's choice not to answer the user's question), and of the response generation component having included information in addition to what the user has requested, i.e. correction of detected invalid plan or suggestion of better alternative (which is again in agreement with CORE). In fact, in these dialog segments, some judges explicitly commented that they found responses in which CORE merely corrected an invalid plan or suggested a better alternative without answering the user's question to be uncooperative. Note that these three dialogs share a common feature in that the user's question can be answered in one short sentence, such as "*Dr. Brown is teaching CS883*" or "*CS*881 *is a natural language processing course*". On the other hand, in dialog IP3, the user asked about how she should go about signing up for independent study (when she was not eligible to do so), the answer to which required enumerating the three steps necessary to carry out the intended action. In this example, CORE's response that corrected the invalid proposal but did not answer the question was ranked highest of all alternative responses, including the response in which CORE answered the question and corrected the proposal. Furthermore, some judges explicitly stated that they found the alternative long answer (describing the procedure for signing up for an independent study when the user is not eligible) to be irrelevant and useless. This suggests that it may not be sufficient to simply adopt a strategy that either always answers the user's question or always ignores the user's question when it is determined to be irrelevant; instead, the judges' evaluation suggests that perhaps a response generation strategy that takes into account the amount of information that needs to be provided to answer an irrelevant question may be more appropriate. However, in any case, the judges agreed that providing additional information to correct invalid plans or to suggest better alternatives are cooperative strategies for a response generation system.

Table 4(b) presents an analysis of the alternative response that the judges liked significantly better than CORE's response in correcting conflicting beliefs; the columns of Table 4(b) show the outcome of the evaluation process (the **Evaluate-Belief** algorithm) and of selecting the rejected belief(s) that the system will explicitly address in its response (the **Select-Focus-Modification** algorithm). Our analysis shows that the preferred response agreed with CORE in rejecting the user's proposed beliefs, but chose to address all four rejected beliefs in the user's proposal individually, while CORE selected one critical piece of proposed evidence to refute. However, in the second dialog segment on correcting conflicts in beliefs (CB2), the top two preferred responses both addressed merely the piece of evidence proposed to support the main belief.† The overall results of our evaluation lead us to conclude that further research is needed to determine the reasons that led the judges to provide seemingly contradictory judgments, and how these factors can be incorporated into CORE's algorithms to improve its performance. Although the best measure of performance would be to evaluate how our response generation strategies contribute to task success within a robust natural language advisement system, which is beyond our current capability, note that CORE's current strategies for correcting invalid proposals, suggesting better alternatives, and resolving conflicts in proposed beliefs result in responses that most of our judges consider concise, coherent and effective and thus provide an excellent basis for future work.

7. Related work

7.1. MODELING COLLABORATION

Allen (1991) and Traum (1993) proposed a discourse model that differentiates among the shared and individual beliefs that agents might hold during collaboration. Their model consists of six plan modalities, organized hierarchically with inheritance in order to accommodate the different states of beliefs during collaboration. The plan modalities include plan fragments that are private to an agent, those proposed by an agent but not yet acknowledged by the other, those proposed by an agent and acknowledged but not yet accepted by the other agent, and a shared plan between the two agents. Plan fragments move from the lower-level modalities (private plans) to the top-level shared plans if appropriate acknowledgment/acceptance is given. Although their framework provides a good basis for representing the state of collaborative planning, it does not specify how the collaborative planning process should be carried out and how responses should be generated when disagreements arise in such planning dialogs.

Grosz and Sidner developed a formal model that specifies the beliefs and intentions that must be held by collaborative agents in order for them to construct a shared plan (Grosz & Sidner, 1990). Their model, dubbed the *SharedPlan* model, eliminates the *master–slave assumption* typically made by plan recognition work prior to their effort.

† These two responses differed in the justification that the system provided in support of its refutation of the proposed evidence. However, this is beyond the scope of this paper and will not be further addressed. Interested readers should refer to Chu-Carroll and Carberry (1998).

Thus, instead of treating collaborative planning as having one controlling agent and one reactive agent where the former has absolute control over the formation of the plan and the latter is involved only in the execution of the plan, they view collaborative planning as "two agents develop[ing] a plan together rather than merely execut[ing] the existing plan of one of them" (Grosz & Sidner, 1990, p. 427). Lochbaum (1994) developed an algorithm for modeling discourse using this SharedPlan model and showed how information-seeking dialogs could be modeled in terms of attempts to satisfy knowledge preconditions (Lochbaum, 1995). Grosz and Kraus (1996) extended the SharedPlan model to handle actions involving groups of agents and complex actions that decompose into multi-agent actions. They proposed a formalism for representing collaborative agents' SharedPlans using three sources of information: (1) the agents' intention to do some actions, (2) their intentions that other agents will carry out some actions and (3) their intention that the joint activity will be successful. However, in their model the agents will avoid adopting conflicting intentions, instead of trying to resolve them.

Sidner analysed multi-agent collaborative planning discourse and formulated an artificial language for modeling such discourse using proposal/acceptance and proposal/rejection sequences (Sidner, 1992, 1994). In other words, a multi-agent collaborative planning process is represented in her language as one agent making a proposal (of a certain action or belief) to the other agents, and the other agents either accepting or rejecting this proposal. Each action (such as *propose* or *accept*) is represented by a message sent from one agent to another, which corresponds to the natural language utterances in collaborative planning discourse. Associated with each message is a set of actions that modifies the stack of open beliefs, rejected beliefs, individual beliefs and mutual beliefs, that facilitate the process of belief revision. However, it was not Sidner's intention to specify conflict detection and resolution strategies for agents involved in collaborative interactions. Walker (1996) also developed a model of collaborative planning in which agents propose options, deliberate on proposals that have been made, and either accept or reject proposals. Our *Propose–Evaluate–Modify* framework builds on these notions of proposal/acceptance and proposal/rejection sequences during collaborative planning.

Walker argues against what she terms the redundancy constraint in discourse (the constraint that redundant information should be omitted) (Walker, 1996). She notes that this constraint erroneously assumes that a hearer will automatically accept claims that are presented to him, and would cause the speaker to believe that it is unnecessary to present evidence that the hearer already knows or should be able to infer (even though this evidence may not currently be part of his attentional focus). Walker investigated the efficiency of different communicative strategies, particularly the use of informationally redundant utterances (IRUs), under different assumptions about resource limits and processing costs and her work suggests that effective use of IRU's can reduce effort during collaborative planning and negotiation.

Heeman and Hirst (1995) investigated collaboration on referring expressions of objects co-present with the dialog participants. They viewed the processes of building referring expressions and identifying their referents as a collaborative activity, and modeled them in a plan-based paradigm. Their model allows for negotiation in selecting amongst multiple candidate referents; however, such negotiation is restricted to the disambiguation process, instead of a negotiation process in which agents try to resolve conflicting beliefs.

Edmonds (1994) studied an aspect of collaboration similar to Heeman. However, he was concerned with collaborating on references to objects that are not mutually known to the dialog participants (such as references to landmarks in direction-giving dialogs). Again, Edmonds captures referent identification as a collaborative process and models it within the planning/plan recognition paradigms. However, he focuses on situations in which an agent's first attempt at describing a referent is considered insufficient by the recipient and the agents collaborate on expanding the description to provide further information and does not consider cases in which conflicts arise between the agents during this process.

Ramshaw (1987) modeled problem-solving actions such as evaluating a plan or considering alternative plans in parallel for the purpose of understanding user utterances. However, he was not concerned with agents collaborating on what problem-solving actions to pursue for the purpose of responding to such user utterances, which is the focus of our work.

Baker (1994) developed a model for negotiation in collaborative learning dialogs, in which two agents successively *refine* each other's offers in order to reach an agreement on a set of propositions. He then analysed the *offer* and *accept* actions that typically make up such negotiation dialogs, specified how application of these actions update the context with respect to both the speaker and the hearer and analysed the relations between successive offers, such as one offer generalizing or subtyping another. Similar to Sidner (1994), Baker focused on the analysis of negotiation dialogs instead of developing mechanisms for participating in such dialogs.

7.2. COOPERATIVE RESPONSE GENERATION

Many researchers (McKeown, Wish & Matthews, 1985; Paris, 1988; McCoy, 1988; Sarner & Carberry, 1990; Zukerman & McConachy, 1993; Logan *et al*., 1994) have argued that information from the user model should affect a generation system's decision on *what* to say and *how* to say it. One user model attribute with such an effect is the user's domain knowledge, which Paris (1988) argues not only influences the *amount* of information given [based on Grice's Maxim of Quantity (Grice, 1975)], but also the *kind* of information provided. McCoy (1988) uses the system's model of the user's domain knowledge to determine possible reasons for a detected misconception and to provide appropriate explanations to correct the misconception. Cawsey (1990) also uses a model of user domain knowledge to determine whether or not a user knows a concept in her tutorial system, and thereby determine whether further explanation is required. Sarner and Carberry (1990) take into account the user's possible plans and goals to help the system determine the user's perspective and provide definitions suitable to the user's needs. McKeown *et al*. (1985) inferred the user's goal from her utterances and tailored the system's response to that particular viewpoint. In addition, Zukerman and McConachy (1993) took into account a user's possible inferences in generating concise discourse.

Logan *et al*., in developing their automated librarian (Cawsey *et al*., 1993; Logan *et al*., 1994), introduced the idea of utilizing a belief revision mechanism (Galliers, 1992) to predict whether a given set of evidence is sufficient to change a user's existing belief. They argued that in the information retrieval dialogs they analysed, "in no cases does negotiation extend beyond the initial belief conflict and its immediate resolution" (Logan *et al.*, 1994, p. 141); thus they do not provide a mechanism for extended collaborative negotiation. On the other hand, our analysis of naturally occurring collaborative negotiation dialogs shows that conflict resolution does extend beyond a single exchange of conflicting beliefs; therefore we employ a recursive *Propose–Evaluate–Modify* framework that allows for extended negotiation. Furthermore, their system deals with one conflict at a time, while our model is capable of selecting a focus in its pursuit of conflict resolution when multiple conflicts arise.

Moore and Paris (1993) developed a text planner that captures both intentional and rhetorical information. Since their system includes a *Persuade* operator for convincing a user to perform an action, it does not assume that the hearer would perform a recommended action without additional motivation. However, although they provide a mechanism for responding to requests for further information, they do not identify strategies for negotiating with the user if the user expresses conflict with the system's recommendation.

Traum and Allen (1994) developed a discourse actor which, among other responsibilities, selects an appropriate system action (sometimes in the form of response generation) based on the current state of the conversational model. However, although their system may choose to deal with a non-yet-accepted proposal, it does not include mechanisms for determining how to go about addressing such a proposal; therefore their system cannot engage in collaborative negotiation for conflict resolution between the system and the user.

Van Beek (1987) developed a system that produces responses by taking into account the overall plans and goals hypothesized for a user, based on Joshi *et al.* (1984). His user goals include desirable characteristics that our system treats as preferences, and he evaluates a plan based on the number of goals it achieves. Thus van Beek's system is unable to reason about how important a particular attribute is to a user or take into consideration the closeness of the matches when it is not possible to exactly satisfy every preference. We argue that this does not accurately model the way humans evaluate alternative plans (Carberry *et al.*, 1999).

8. Future work

There are several directions in which the work described in this paper can be extended. First of all, discussions in this paper have been limited to collaborative planning scenarios in which two agents work together to achieve a shared goal. Intutively, the *Propose–Evaluate–Modify* framework developed to model such collaborative planning activities would seem to be applicable to modeling collaborative planning among multiple (three or more) agents. However, problems arise when we consider situations in which more than one agent rejects a proposal and attempts to propose a modification to the proposal. Since the agents have different domain knowledge and preferences, they may propose to modify the proposal in different manners. Two important problems need to be addressed in order for the agents to continue their collaborative planning in such a scenario. The first problem involves the coordination of activities among multiple collaborative agents, namely, how should dialog turn-taking among the participants be controlled in order to prevent multiple agents from proposing modifications to propo-

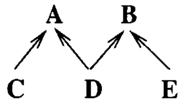


FIGURE 14. Example of a belief playing multiple roles.

sals at the same time? If a strict ordering scheme should be adopted, what should this scheme be? The second problem involves the evaluation and modification strategies that an agent should employ when participating in multi-agent collaborative planning. Suppose that an agent rejects a proposal and, before she proposes her modification, another agent proposes a different modification to the proposal. Should the first agent now propose her original modifications as if the second agent had not made a proposal? Should she evaluate the second agent's proposal of modification and perhaps accept it instead of making her own proposal? Or should she attempt to come up with a new proposal of modification that satisfies the intentions of both agents if possible? In order to select among these alternatives, a more sophisticated decision-making mechanism than that employed in two-agent collaborative planning must be developed.

An important assumption made in this paper regarding the relationships between proposed beliefs is that proposed beliefs can always be represented in a tree structure, i.e. each time a belief is proposed, it is intended as support for only one other belief. Relaxing this assumption complicates the selection of the focus of modification during the conflict resolution process. For instance, consider the proposed belief structure in Figure 14. Suppose that the system evaluates the proposal and rejects all proposed beliefs A, B, C, D, and E. In selecting the focus of modification, should the system now prefer addressing D because its resolution will potentially resolve the conflict about both A and B? What if D is the belief which the system has the least amount of evidence against? Further research is needed to determine how the current algorithms for conflict resolution should be modified to accommodate such belief structures.

Finally, in this paper we have focused on the content selection process in response generation. For text structuring, we used the simple strategy of presenting claims before their justification. However, Cohen (1987) analysed argumentative texts and found variations in the order in which claims and their evidence are presented. Since text structure can influence coherence and focus, we must investigate appropriate mechanisms for determining the structure of a response containing multiple propositions. In addition, we must identify appropriate syntactic forms for expressing each utterance (such as a surface negative question versus a declarative statement), identify when cue words should be employed, and use a sentence realizer to produce actual English utterances.

9. Conclusions

In order for an agent to successfully collaborate with other agents in developing a plan to achieve a goal, she must be capable of dealing with situations in which conflicts occur. Conflict detection requires that an agent evaluate proposals based on her private beliefs, while conflict resolution involves communication among agents for the purpose of squaring away the detected conflicts. This paper has presented a plan-based model that specifies how a collaborative agent should detect and attempt to resolve conflicts that arise during collaborative planning. Our model provides an overall framework for modelling such activity by capturing collaborative activities in a recursive *Propose–Evaluate–Modify* cycle of actions. Instead of slavishly accepting proposals made by the executing agent, our system evaluates a proposal based on its private beliefs about the domain and about the executing agent in order to determine the validity and optimality of the proposed plan and the truth of the proposed beliefs. In situations where a relevant conflict is detected, our system initiates a collaborative negotiation subdialog in an attempt to resolve the conflict. Our mechanism for conflict resolution identifies, in cases where multiple conflicting beliefs exist, the focus of modification based on its predicted success in resolving the conflict about the rejected top-level proposed belief, thus leading the agent to engage in effective and efficient dialogs. In cases where an agent's first attempt to resolve a conflict fails, our model will allow the agent to apply the same conflict detection and resolution methods to the other agent's new proposal, resulting in an embedded negotiation subdialog captured by recursive meta-plans. In addition, our model accounts for why questions are sometimes never answered even in the most cooperative of environments.

The authors would like to thank Candy Sidner, Stephanie Elzer, Kathy McCoy, Lance Ramshaw, and Suzanne Mannes for their helpful comments and discussions, David Traum and the anonymous reviewers for their comments on an earlier draft of this paper, and Rachel Sacher for her help in the implementation of CORE. This material is based upon work supported by the National Science Foundation under Grant No. IRI-9122026.

References

ALLEN, J. (1991). Discourse structure in the TRAINS project. In *Darpa Speech and Natural Language Workshop*, pp. 325–330.

BAKER, M. (1994). A model for negotiation in teaching–learning dialogs. *Journal of Artifical Intelligence in Education*, **5,** 199–254.

BIRNBAUM, L., FLOWERS, M. & MCGUIRE, R. (1980). Towards an AI model of argumentation. In *Proceedings of the National Conference on Artificial Intelligence*, pp. 313–315.

CARBERRY, S., CHU-CARROLL, J. & ELZER S. (1999). Constructing and utilizing a model of user preferences in collaborative consultation dialogs. *Computational Intelligence*, **15,** 185–217.

CAWSEY, A., GALLIERS, J., LOGAN, B., REECE, S. & SPARCK JONES, K. (1993). Revising beliefs and intentions: a unified framework for agent interaction. In *The Ninth Biennial Conference of the Society for the Study of Artificial Intelligence and Simulation of Behaviour*, pp. 130–139.

CAWSEY, A. (1990). Generating explanatory discourse. In R. DALE, C. MELLISH & M. ZOCK, Eds. *Current Research in Natural Language Generation*, Chapter 4, pp. 75–101. New York: Academic Press.

CAWSEY, A. (1993). *Explanation and Interaction.* New York: MIT Press.

CHU-CARROLL, J. & CARBERRY, S. (1994). A plan-based model for response generation in collaborative task-oriented dialogs. In *Proceedings of the 12th National Conference on Artificial Intelligence*, Seattle, WA, pp. 799–805.

CHU-CARROLL, J. & CARBERRY, S. (1995*a*) Communication for conflict resolution in multi-agent collaborative planning. In *Proceedings of the First International Conference on Multiagent Systems*, San Francisco, CA. pp. 49–56.

CHU-CARROLL, J. & CARBERRY, S. (1995*b*). Generating information-sharing subdialogs in expert-user consultation. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, Montreal, Canada. pp. 1243–1250.

CHU-CARROLL, J. & CARBERRY, S. (1995*c*). Response generation in collaborative negotiation. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, Boston, MA. 136–143.

CHU-CARROLL, J. & CARBERRY, S. (1996). Conflict detection and resolution in collaborative planning. In WOOLDRIDGE, MULLER & TAMBE, Eds. *Intelligent Agents: Agent Theories, Architectures, and Languages, Vol. II*, Lecture Notes, pp. 111–126. Berlin: Springer-Verlag.

CHU-CARROLL, J. & CARBERRY, S. (1998). Collaborative response generation in planning dialogs. *Computational Linguistics*, **24,** 355–400.

CHU-CARROLL, J. (1996). *A plan-based model for response generation in collaborative consultation dialogs*. Ph.D. Thesis, University of Delaware. Also available as Department of Computer and Information Sciences, Laboratories for NLP/AI/HCI, Technical Report 97-01.

CLARK, H. H. & SCHAEFER, E. F. (1989). Contributing to discourse. *Cognitive Science*, **13,** 259–294.

CLARK, H. & WILKES-GIBBS, D. (1990). Referring as a collaborative process. In P. COHEN, J. MORGAN & M. POLLACK, Eds. *Intentions in Communication*, Chapter 23, pp. 463–493. New York: MIT Press.

COHEN, P. R. (1985). *Heuristic Reasoning about Uncertainty: An Artificial Intelligence Approach.* London: Pitman Publishing Company.

COHEN, R. (1987). Analyzing the structure of argumentative discourse. *Computational Linguistics*, **13,** 11–24.

COLUMBIA UNIVERSITY TRANSCRIPTS (1985). Transcripts derived from audiotape conversations made at Columbia University, New York, NY. Provided by Kathleen McKeown.

EDMONDS, P. G. (1994). Collaboration on reference to objects that are not mutually known. In *Proceedings of the 15th International Conference on Computational Linguistics*, pp. 1118–1122.

ELLER, R. & CARBERRY, S. (1992). A meta-rule approach to flexible plan recognition in dialog. *User Modeling and User-Adapted Interaction*, **2,** 27–53.

ELZER, S., CHU-CARROLL, J. & CARBERRY, S. (1994). Recognizing and utilizing user preferences in collaborative consultation dialogs. In *Proceedings of the 4th International Conference on User Modeling*, Hyannis, MA. pp. 19–24.

FLOWERS, M. & DYER, M. (1984). Really arguing with your computer. In *Proceedings of the National Computer Conference*, pp. 653–659.

GALLIERS, J. R. (1992). Autonomous belief revision and communication. In GARDENFORS, Ed. *Belief Revision.* Cambridge: Cambridge University Press.

GRICE, H. P. (1975). Logic and conversation. In P. COLE & J. L. MORGAN, Eds. *Syntax and Semantics 3: Speech Acts*, pp. 41–58. New York: Academic Press, Inc.

GROSS, D., ALLEN, J. F. & TRAUM, D. R. (1993). *The TRAINS 91 dialogs.* Technical Report TN92-1, Department of Computer Science, University of Rochester.

GROSZ, B. & KRAUS, S. (1996). Collaborative plans for complex group actions. *Artificial Intelligence*, **86,** 269–357.

GROSZ, B. J. & SIDNER, C. L. (1990). Plans for discourse. In COHEN, MORGAN, & POLLACK, Eds. *Intentions in Communication*, Chapter 20, pp. 417–444. New York: MIT Press.

HEEMAN, P. A. & HIRST, G. (1995). Collaborating on referring expressions. *Computational Linguistics*, **21,** 351–382.

JOSHI, A., WEBBER, B. & WEISCHEDEL, R. M. (1984). Living up to expectations: Computing expert responses. In *Proceedings of the Fourth National Conference on Artificial Intelligence*, Austin, TX. pp. 169–175.

KEENEY, R. L. & RAIFFA, H. (1993). *Decisions with Multiple Objectives: Preferences and Value Tradeoffs.* Cambridge: Cambridge University Press.

LAMBERT, L. & CARBERRY, S. (1991). A tripartite plan-based model of dialog. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, Berkeley, CA. pp. 47–54.

LAMBERT, L. & CARBERRY, S. (1992). Modeling negotiation dialogs. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, Newark, DE. pp. 193–200.

LOCHBAUM, K. E. (1994). *Using collaborative plans to model the intentional structure of discourse.* Ph.D. Thesis, Harvard University.

LOCHBAUM, K. E. (1995). The use of knowledge preconditions in language processing. In *Proceedings of the International Joint Conference on Artificial Intelligence*, Montreal, Canada. pp. 1260–1266.

LOGAN, B., REECE, S., CAWSEY, A., GALLIERS, J. & SPARCK JONES, K. (1994). *Belief revision and dialog management in information retrieval*. Technical Report 339, University of Cambridge, Computer Laboratory.

MAYBURY, M. T. (1992). Communicative acts for explanation generation. *International Journal of Man-Machine Studies*, **37,** 135–172.

MAYBURY, M. T. (1993). Communicative acts for generating natural language arguments. In *Proceedings of the National Conference on Artificial Intelligence*, pp. 357–364.

MCCOY, K. F. (1988). Reasoning on a highlighted user model to respond to misconceptions. *Computational Linguistics*, **14,** 52–63.

MCKEOWN, K. R., WISH, M. & MATTHEWS, K. (1985). Tailoring explanations for the user. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, pp. 794–798. Los Angeles, CA.

MCKEOWN, R. (1985). *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*. Cambridge: Cambridge University Press.

MOORE, J. D., PARIS, C. L. (1993). Planning text for advisory dialogs: capturing intentional and rhetorical information. *Computational Linguistics*, **19,** 651–694.

PARIS, C. L. (1998). Tailoring object descriptions to a user's level of expertise. *Computational Linguistics*, **14,** 64–78.

PAYNE, J. W., BETTMAN, J. R. & JOHNSON, E. J. (1993). *The Adaptive Decision Maker*. Cambridge: Cambridge University Press.

POLLACK, M. E. (1986). A model of plan inference that distinguishes between the beliefs of actors and observers. In *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics*, pp. 207–214.

QUILICI, A. (1992). Arguing about planning alternatives. In *Proceedings of the 14th International Conference on Computational Linguistics*, pp. 906–910.

RAMSHAW, L. A. (1987). *Pragmatic knowledge for resolving ill-formedness*. Ph.D. Thesis, University of Delaware.

RASKUTTI, B. & ZUKERMAN, I. (1994). Query and response generation during information-seeking interactions. In *Proceedings of the 4th International Conference on User Modeling*, pp. 25–30.

REICHMAN, R. (1981). Modeling informal debates. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, pp. 19–24.

ROSENSCHEIN, J. S. & ZLOTKIN, G. (1994). *Rules of Encounter—Designing Conventions for Automated Negotiation among Computers*. New York: MIT Press.

SARNER, M. H. & CARBERRY, S. (1990). Tailoring explanations using a multifaceted user model. In *Proceedings of the Second International Workshop on User Models*, Honolulu, Hawaii, March.

SENEFF, S., HIRSCHMAN, L. & ZUE, V. (1991). Interactive problem solving and dialog in the ATIS domain. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pp. 354–359.

SIDNER, C. L. (1992). Using discourse to negotiate in collaborative activity: an artificial language. In *AAAI-92 Workshop: Cooperation Among Heterogeneous Intelligent Systems*, San Jose, CA. pp. 121–128.

SIDNER, C. L. (1994). An artificial discourse language for collaborative negotiation. In *Proceedings of the 12th National Conference on Artificial Intelligence*, Seattle, WA. pp. 814–819.

SRI Transcripts (1992). Transcripts derived from audiotape conversations made at SRI International, Menlo Park, CA. Prepared by Jacqueline Kowtko under the direction of Patti Price.

SYCARA, K. (1989). Argumentation: planning other agents' plans. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, pp. 517–523.

TRAUM, D. R. & ALLEN, J. F. (1994). Discourse obligations in dialog processing. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, Las Cruces, NM. pp. 1–8.

TRAUM, D. R. (1993). Mental state in the TRAINS-92 dialog manager. In *Working Notes of the AAAI Spring Symposium on Reasoning about Mental States: Formal Theories and Applications*, pp. 143–149.

UDEL TRANSCRIPTS (1995). Transcripts derived from audiotape conversations made at the University of Delaware. Recorded and transcribed by Rachel Sacher.

VAN BEEK, P., COHEN, R. & SCHMIDT, K. (1993). From plan critiquing to clarification dialog for cooperative response generation. *Computational Intelligence*, **9,** 132–154.

van Beek, P. G. (1987). A model for generating better explanations. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*, pp. 215–220. Stanford, CA.

Walker, M. & Whittaker, S. (1990). Mixed initiative in dialog: an investigation into discourse segmentation. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, Pittsburgh, PA. pp. 70–78.

Walker, M. A. (1992). Redundancy in collaborative dialog. In *Proceedings of the 15th International Conference on Computational Linguistics*, pp. 345–351.

Walker, M. A. (1996). The effect of resource limits and task complexity on collaborative planning in dialog. *Artificial Intelligence*, **85,** 181–243.

Whittaker, S. & Stenton, P. (1988). Cues and control in expert-client dialogs. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, Buffalo, NY. pp. 123–130.

Young, R. M., Moore, J. D. & Pollack, M. E. (1994). Towards a principled representation of discourse plans. In *Proceedings of the Sixteenth Annual Meeting of the Cognitive Science Society*, pp. 946–951.

Zukerman, I. & McConachy, R. (1993). Generating concise discourse that addresses a user's inferences. In *Proceedings of the 1993 International Joint Conference on Artificial Intelligence*.

Appendix A: sample dialog in evaluation questionnaire

This section presents a sample dialog from the questionnaire used for the evaluation of CORE, as discussed in Section 6.2. The dialog is annotated to indicate CORE's actual response to the user's utterance, as well as how CORE's response generation algorithm is modified to obtain each alternative response. These annotations are included as comments (surrounded by /* and */) and were not available to the judges during the evaluation process.

Question: /* *This dialog corresponds to IP2 in Section 6.2.* */

```
U: I want to satisfy my seminar course requirement in CS.
   What is the content of CS881?
```

Here the user wants to take CS881 as part of satisfying her seminar course requirement. However, CORE believes that CS881 is not offered this semester. Now consider the following alternative responses:

Response 1: /* *CORE does not evaluate the validity of the user proposal, but simply provides a direct answer to the user's question* */

```
CORE: CS881 is a natural language processing course.
```

Response 2: /* *CORE points out the invalid action, but does not provide evidence to support the claim* */

```
CORE: You cannot take CS881.
```

Response 3: /* *CORE's actual response* */

```
CORE: You cannot take CS881.
      CS881 is not offered this semester.
```

Response 4: /* *CORE answers user question, points out invalid action, and provides evidence to support it. This is the preferred response* $IP2$ *in Table 4*(*a*) */

```
CORE: CS881 is a natural language processing course.
      However, you cannot take CS881.
      CS881 is not offered this semester.
```

Response 5: /* *CORE does not explicitly point out the invalid action, but instead presents evidence in order for the user to infer the invalidity* */

```
CORE: CS881 is not offered this semester.
```