

# RESPONDING TO USER QUERIES IN A COLLABORATIVE ENVIRONMENT\*

Jennifer Chu

Department of Computer and Information Sciences  
University of Delaware  
Newark, DE 19716, USA  
Internet: jchu@cis.udel.edu

## Abstract

We propose a plan-based approach for responding to user queries in a collaborative environment. We argue that in such an environment, the system should not accept the user's query automatically, but should consider it a proposal open for negotiation. In this paper we concentrate on cases in which the system and user disagree, and discuss how this disagreement can be detected, negotiated, and how final modifications should be made to the existing plan.

## 1 Introduction

In task-oriented consultation dialogues, the user and expert jointly construct a plan for achieving the user's goal. In such an environment, it is important that the agents agree on the domain plan being constructed and on the problem-solving actions being taken to develop it. This suggests that the participants communicate their disagreements when they arise lest the agents work on developing different plans. We are extending the dialogue understanding system in [6] to include a system that responds to the user's utterances in a collaborative manner.

Each utterance by a participant constitutes a *proposal* intended to affect the agents' shared plan. One component of our architecture, the *evaluator*, examines the user's proposal and decides whether to accept or reject it. Since the user has knowledge about his/her particular circumstances and preferences that influence the domain plan and how it is constructed, the evaluator must be a *reactive planner* that interacts with the user to obtain information used in building the evaluation meta-plan. Depending on the evaluation, the system can accept or reject the proposal, or suggest what it considers to be a better alternative, leading to an embedded negotiation subdialogue.

In addition to the evaluator, our architecture consists of a *goal selector*, an *intentional planner*, and a *discourse realizer*. The goal selector, based on the result of the evaluation and the current dialogue model, selects an appropriate intentional goal for the system to pursue. The intentional planner builds a plan to achieve the intentional goal, and the discourse realizer generates utterances to convey information based on the intentional plan.

This paper describes the evaluator, concentrating on cases in which the system and user disagree. We show how the system determines that the user's proposed additions are erroneous and, instead of directly responding to the user's utterances, conveys the disagreement. Thus, our work contributes to an overall dialogue system by 1) extending the model in [6] to eliminate the assumption that the system will automatically answer the user's questions or follow the user's proposals, and 2) capturing the notion

of cooperative responses within an overall collaborative framework that allows for negotiation.

## 2 The Tripartite Model

Lambert and Carberry proposed a plan-based tripartite model of expert/novice consultation dialogue which includes a *domain* level, a *problem-solving* level, and a *discourse* level [6]. The domain level represents the system's beliefs about the user's plan for achieving some goal in the application domain. The problem-solving level encodes the system's beliefs about how both agents are going about constructing the domain plan. The discourse level represents the system's beliefs about both agents' communicative actions. Lambert developed a plan recognition algorithm that uses contextual knowledge, world knowledge, linguistic clues, and a library of generic recipes for actions to analyze utterances and construct a dialogue model[6].

Lambert's system automatically adds to the dialogue model all actions inferred from an utterance. However, we argue that in a collaborative environment, the system should only accept the proposed additions if the system believes that they are appropriate. Hence, we separate the dialogue model into an *existing* dialogue model and a *proposed* model, where the former constitutes the *shared plan* agreed upon by both agents, and the latter the newly proposed actions that have not yet been confirmed.

Suppose earlier dialogue suggests that the user has the goal of getting a Master's degree in CS (*Get-Masters(U,CS)*). Figure 1 illustrates the dialogue model that would be built after the following utterances by Lambert's plan recognition algorithm modified to accommodate the separation of the existing and proposed dialogue models, and augmented with a relaxation algorithm to recognize ill-formed plans[2].

*U: I want to satisfy my seminar course requirement.  
Who's teaching CS689?*

## 3 The Evaluator

A collaborative system should only incorporate proposed actions into an existing plan if they are considered appropriate. This decision is made by the *evaluator*, which will be discussed in this section. This paper only considers cases in which the user's proposal contains an infeasible action (one that cannot be performed) or would result in an ill-formed plan (one whose actions do not contribute to one another as intended)[9].

We argue that the evaluator, in order to check for erroneous plans/goals, only needs to examine actions in the *proposed* model, since actions in the existing model would have been checked when they were proposed. When a chain of actions is proposed, the evaluator starts examining from the top-most action so that the most *general* action that is inappropriate will be addressed.

\*This material is based upon work supported by the National Science Foundation under Grant No. IRI-9122026.

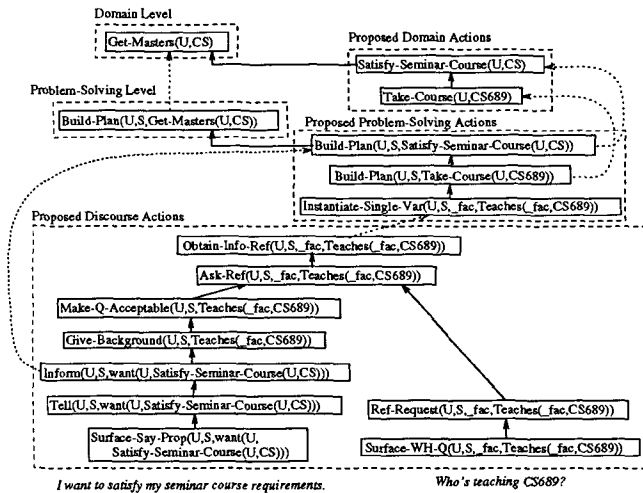


Figure 1: The Structure of the User's Utterances

The evaluator checks whether the existing and proposed actions together constitute a well-formed plan, one in which the children of each action contribute to their parent action. Therefore, for each pair of actions, the evaluator checks against its recipe library to determine if their parent-child relationship holds. The evaluator also checks whether each additional action is feasible by examining whether its applicability conditions are satisfied and its preconditions<sup>1</sup> can be satisfied.

We contend that well-formedness should be checked before feasibility since the feasibility of an action that does not contribute to its parent action is irrelevant. Similarly, the well-formedness of a plan that attempts to achieve an infeasible goal is also irrelevant. Therefore, we argue that the processes of checking well-formedness and feasibility should be *interleaved* in order to address the most general action that is inappropriate. We show how this interleaved process works by referring back to figure 1.

Suppose the system believes that CS689 is not a seminar course. The evaluation process starts from *Satisfy-Seminar-Course(U,CS)*, the top-most action in the proposed domain model. The system's knowledge indicates that *Satisfy-Seminar-Course(U,CS)* contributes to *Get-Masters(U,CS)*. The system also believes that the applicability conditions and the preconditions for the *Satisfy-Seminar-Course* domain plan are satisfied, indicating that the action is feasible. However, the system's recipe library gives no reason to believe that *Take-Course(U,CS689)* contributes to *Satisfy-Seminar-Course(U,CS)*, since CS689 is *not* a seminar course. The evaluator then decides that this pair of proposed actions would make the domain plan ill-formed.

#### 4 When the Proposal is Erroneous

The goal selector's task is to determine, based on the current dialogue model, an *intentional* goal [8] that is most appropriate for the system to pursue. An intentional goal could be to directly respond to the user's utterance,

<sup>1</sup>Both applicability conditions and preconditions are prerequisites for executing a recipe. However, it is unreasonable to attempt to satisfy an applicability condition whereas preconditions can be planned for.

Action:	Correct-Inference(_s1,_s2,_proposed)
Recipe-Type:	Decomposition
Appl Cond:	believe(_s1, ¬contributes(_act1,_act2)) believe(_s2, contributes(_act1,_act2))
Constraints:	in-plan(_act1,_proposed) ∨ in-plan(_act2,_proposed)
Body:	Modify-Acts(_s1,_s2,_proposed,_act1,_act2) Insert-Correction(_s1,_s2,_proposed)
Effects:	modified(_proposed)
Goal:	well-formed(_proposed)
Action:	Modify-Acts(_s1,_s2,_proposed,_act1,_act2)
Recipe-Type:	Specialization
Appl Cond:	believe(_s1, ¬contributes(_act1,_act2))
Preconditions:	believe(_s2, ¬contributes(_act1,_act2))
Body:	Remove-Act(_s1,_s2,_proposed,_act1) Alter-Act(_s1,_s2,_proposed,_act1)
Effects:	modified(_proposed)
Goal:	modified(_proposed)

Figure 2: Two Problem-Solving Recipes

to correct a user's misconception, to provide a better alternative, etc. In this paper we only discuss the goal selector's task when the user has an erroneous plan/goal.

In a collaborative environment, if the system decides that the proposed model is infeasible/ill-formed, it should refuse to accept the additions and suggest modifications to the proposal by entering a negotiation subdialogue. For this purpose, we developed recipes for two problem-solving actions, *Correct-Goal* and *Correct-Inference*, each a specialization of a *Modify-Proposal* action. We illustrate the *Correct-Inference* action in more detail.

We show two problem-solving recipes, *Correct-Inference* and *Modify-Acts*, in figure 2. The *Correct-Inference* recipe is applicable when *\_s2* believes that *\_act1* contributes to achieving *\_act2*, while *\_s1* believes that such a relationship does not hold. The goal is to make the resultant plan a well-formed plan; therefore, its body consists of an action *Modify-Acts* that deletes the problematic components of the plan, and *Insert-Correction*, that inserts new actions/variables into the plan. One precondition in *Modify-Acts* is *believe(\_s2, ¬contributes(\_act1,\_act2))* (note that in *Correct-Inference*, *\_s2* believes *contributes(\_act1,\_act2)*), and the change in *\_s2*'s belief can be accomplished by invoking the discourse level action *Inform* so that *\_s1* can convey the ill-formedness to *\_s2*. This *Inform* act may lead to further negotiation about whether *\_act1* contributes to *\_act2*. Only when *\_s1* receives a positive feedback from *\_s2*, indicating that *\_s2* accepts *\_s1*'s belief, can *\_s1* assume that the proposed actions can be modified.

Earlier discussion shows that the proposed actions in figure 1 would make the domain plan ill-formed. Therefore, the goal selector posts a goal to modify the proposal, which causes the *Correct-Inference* recipe in figure 2 to be selected. The variables *\_act1* and *\_act2* are bound to *Take-Course(U,CS689)* and *Satisfy-Seminar-Course(U,CS)*, respectively, since the system believes that the former does not contribute to the latter.

Figure 3 shows how we envision the planner to expand on the *Correct-Inference* recipe, which results in the generation of the following two utterances:

- (1) *S: Taking CS689 does not contribute to satisfying the seminar course requirement,*
- (2) *CS689 is not a seminar course.*

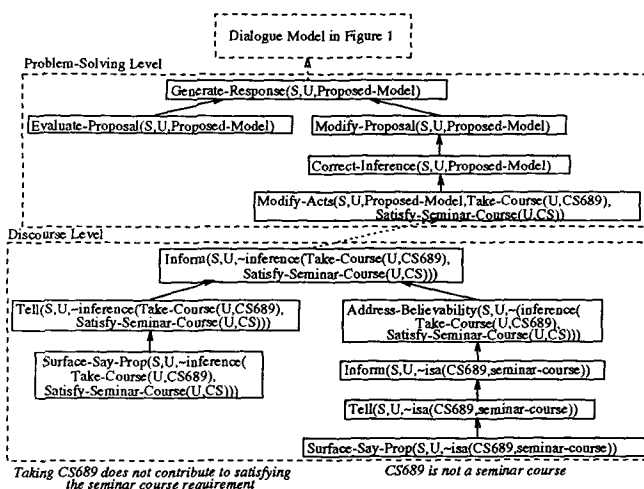


Figure 3: The Dialogue Model for the System's Response

The action  $Inform(s1, s2, prop)$  has the goal  $believe(s2, prop)$ ; therefore, utterance (1) is generated by executing the Inform action as an attempt to satisfy the preconditions for the Modify-Acts recipe. Utterance (2) results from the Address-Believability action, which is a subaction of Inform, to support the claim in (1). The problem-solving and discourse levels in figure 3 operate on the entire dialogue model shown in figure 1, since the evaluation process acts upon this model. Due to this nature, the evaluation process can be viewed as a *meta-planning* process, and when the goal of this process is achieved, the modified dialogue model is returned to.

Now consider the case in which the user continues by accepting utterances (1) and (2), which satisfies the precondition of Modify-Acts. Modify-Acts has two specializations, *Remove-Act*, which removes the incorrect action (and all of its children), and *Alter-Act*, which generalizes the proposed action so that the plan will be well-formed. Since Take-Course contributes to Satisfy-Seminar-Course as long as the course is a seminar course, the system generalizes the user's proposed action by replacing CS689 with a variable. This variable may be instantiated by the *Insert-Correction* subaction of Correct-Inference when the dialogue continues. Note that our model accounts for why the user's original question about the instructor of CS689 is never answered — a conflict was detected that made the question superfluous.

## 5 Related Work

Several researchers have studied collaboration [1, 3, 10] and Allen proposed different plan modalities depending on whether a plan fragment is shared, proposed and acknowledged, or merely private [1]. However, they have emphasized discourse analysis and none has provided a plan-based framework for *proposal negotiation*, specified appropriate system response during collaboration, or accounted for why a question might never be answered. Litman and Allen used discourse meta-plans to handle a class of correction subdialogues [7]. However, their Correct-Plan only addressed cases in which an agent adds a repair step to a pre-existing plan that does not execute as expected. Thus their meta-plans do not handle correction of *proposed* additions to the dialogue model (since this

generally does not involve adding a step to the proposal). Furthermore, they were only concerned with understanding utterances, not with generating appropriate responses. The work in [5, 11, 9] addressed generating cooperative responses and responding to plan-based misconceptions, but did not capture these within an overall collaborative system that must negotiate proposals with the user. Heeman [4] used meta-plans to account for collaboration on referring expressions. We have addressed collaboration in constructing the user's task-related plan, captured cooperative responses and negotiation of how the plan should be constructed, and provided an accounting for why a user's question may never be answered.

## 6 Conclusions and Future Work

We have presented a plan-based framework for generating responses in a collaborative environment. Our framework improves upon previous ones in that, 1) it captures cooperative responses as a part of collaboration, 2) it is capable of initiating negotiation subdialogues to determine what actions should be added to the shared plan, 3) the correction process, instead of merely pointing out problematic plans/goals to the user, modifies the plan into its most specific form accepted by both participants, and 4) the evaluation/correction process operates at a meta-level which keeps the negotiation subdialogue separate from the original dialogue model, while allowing the same plan-inference mechanism to be used at both levels.

We intend to enhance our evaluator so that it also recognizes *sub-optimal* solutions and can suggest better alternatives. We will also study the goal selector's task when the user's plan/goal is well-formed/feasible. This includes identifying a set of *intentional goals* and a strategy for the goal selector to choose amongst them. Furthermore, we need to develop the intentional planner which constructs a plan to achieve the posted goal, and a discourse realizer to generate natural language text.

## References

- [1] James Allen. Discourse structure in the TRAINS project. In *Darpa Speech and Natural Language Workshop*, 1991.
- [2] Rhonda Eller and Sandra Carberry. A meta-rule approach to flexible plan recognition in dialogue. *User Modeling and User-Adapted Interaction*, 2:27--53, 1992.
- [3] Barbara Grosz and Candace Sidner. Plans for discourse. In Cohen et al., editor, *Intentions in Communication*, pages 417--444. 1990.
- [4] Peter Heeman. A computational model of collaboration on referring expressions. Master's thesis, University of Toronto, 1991.
- [5] Aravind Joshi, Bonnie Webber, and Ralph Weischedel. Living up to expectations: Computing expert responses. In *Proc. AAAI*, pages 169--175, 1984.
- [6] Lynn Lambert and Sandra Carberry. A tripartite plan-based model of dialogue. In *Proc. ACL*, pages 47--54, 1991.
- [7] Diane Litman and James Allen. A plan recognition model for subdialogues in conversation. *Cognitive Science*, 11:163--200, 1987.
- [8] Johanna Moore and Cecile Paris. Planning text for advisory dialogues. In *Proc. ACL*, pages 203--211, 1989.
- [9] Martha Pollack. A model of plan inference that distinguishes between the beliefs of actors and observers. In *Proc. ACL*, pages 207--214, 1986.
- [10] Candace Sidner. Using discourse to negotiate in collaborative activity: An artificial language. In *Workshop Notes: AAAI-92 Cooperation Among Heterogeneous Intelligent Systems*, pages 121--128, 1992.
- [11] Peter van Beek. A model for generating better explanations. In *Proc. ACL*, pages 215--220, 1987.