Recognizing and Utilizing User Preferences in Collaborative Consultation Dialogues*

Stephanie Elzer
Department of Computer Science
University of Delaware
Newark, DE 19716, USA
Email: elzer@cis.udel.edu

Jennifer Chu-Carroll
Department of Computer Science
University of Delaware
Newark, DE 19716, USA
Email: jchu@cis.udel.edu

Sandra Carberry
Department of Computer Science
University of Delaware
Visitor: IRCS
University of Pennsylvania
Email: carberry@cis.udel.edu

Abstract

A natural language collaborative consultation system must take user preferences into account. This paper presents two strategies: one for dynamically recognizing user preferences during the course of a collaborative planning dialogue and another for exploiting the model of user preferences to detect suboptimal solutions and suggest better alternatives. By modeling and utilizing user preferences, our system is better able to fulfill its role as a collaborative agent.

Introduction

We have been developing a natural language consultation system in which the system plays the role of an expert in the domain and collaborates with the user to construct a plan for achieving the user's domain goals. Many of these domains involve preferential choice. That is, there are a wide variety of individual preferences that determine which solution will be considered best by a particular user. Kass and Finin classify these preferences under the category of attitudes, and observe that people ''exhibit preferences and bias toward particular options or solutions. A natural language system may often need to recognize the bias and preferences a user has in order to communicate effectively.'' (Kass & Finin 1988)

This observation is particularly relevant to collaborative consultation dialogues because of the different types of knowledge that the system and user bring to the dialogue. The system presumably has more extensive and accurate domain knowledge than the user, while the user has knowledge about his particular circumstances, intentions and preferences that are either restrictions on or potential influencers (Bratman 1990) of the domain plan that is constructed. If the domain involves preferential choice, the system needs to be able to evaluate potential solutions with respect to the user's preferences in order to determine when one set of actions is more desirable than another. Thus the system must be able to recognize and make use of the user's preferences in order to fulfill its role as a collaborative agent.

In this paper, we present two strategies: one for dynamically recognizing user preferences during the course of a collaborative planning dialogue and another for exploiting the model of user preferences to detect suboptimal solutions and suggest better alternatives. Our recognition strategy is the first to recognize preferences during the course of a natural dialogue, utilizing characteristics of both the utterance itself and the dialogue in developing a model of user preferences. Our generation strategy reflects an additive model of human decision-making (Reed 1982) and takes into account both the strength of a preference and the closeness of a potential match. The examples in this paper have been run on our implemented system and are taken from a university domain of degrees, courses and requirements.

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

Preferences Versus Goals

One might argue that user preferences could be represented as user goals (van Beek 1987). However, preferences and goals play substantially different roles in plan construction. Users are conscious of their goals and have the intention of satisfying them. Therefore, in helping the user develop a plan, the system uses the user's goals to guide the planning process and to limit the search for a plan of action. Preferences, on the other hand, influence the selection and instantiation of the actions which achieve the user's goals.

Although an agent may relinquish some of his goals if the entire goal set cannot be satisfied, the intent at the outset is to develop a plan that achieves all of the user's goals. If preferences were treated as goals, actions might be added to the plan being developed in order to fulfill a preference. Suppose, for example, that the user has the goal of satisfying his foreign language requirement and a preference for morning courses, which cannot both be satisfied by a single *Take-Course* action. If the preference is treated as a goal, the attempt to satisfy all existing goals will lead the system to develop a plan containing two *Take-Course* actions. In all likelihood, this is not what the user wants — the user is only interested in taking courses which meet in the morning with respect to the goal of satisfying the foreign language requirement. In addition, users are often not conscious of their preferences at the outset of planning and only bring these preferences into play as they must evaluate alternative actions and choose among them. Thus, we contend that preferences should not be treated as goals, but as potential influences (Bratman 1990) held with respect to actions.

Overview

We are currently modeling *attribute-value preferences*, where an attribute-value preference is a preference that an object in an action have a particular attribute value. That is, if the user wants to execute an action *Take-Course(U, _crs)*[1], the user might have a preference for courses that meet in the morning and a preference for courses taught by a certain professor. We are currently making the strong assumption that attribute preferences are independent of one another. For example, we do not consider situations in which a user's preference regarding the meeting time of a course is dependent on the type of course (music, math, etc.)

Our system recognizes both preferences which are expressed by the user and preferences which can be detected through patterns in the user's acceptance or rejection of proposals. For example, a user may reject several solutions which have common characteristics without ever explicitly expressing a preference. By reasoning about this pattern of rejection, our system can detect a possible user preference that is only implicitly conveyed.

People often weigh their preferences when making a decision, and consider some preferences to be more important than others. Thus a complete representation of user preferences must capture the strength of each preference. Our system deduces the strength of a preference based on a combination of the conversational circumstances in which the preference is expressed and the semantics of the user's utterance. In addition, a consultant should have more confidence in some preferences that have been deduced than in others. For example, a consultant will have more confidence in beliefs about an agent's preferences if they result from the agent's explicit statement to that effect than if they are deduced from a pattern of rejections by the agent. Our system captures this by using endorsements (Cohen 1985) to reflect the reliability of evidence. It is important to note the difference between the strength of a preference and the strength of an endorsement of the preference: the strength of the preference represents how important the preference is to the user while the strength of the endorsement represents how strongly the system believes that the preference accurately models an attitude of the user.

Our system utilizes the recognized preferences to suggest better alternatives when an action is proposed by the user. We have developed a **ranking advisor** for rating different instantiations of an action based on the user's preferences. The ranking advisor selects preferences that are sufficiently reliable (as indicated by their endorsements) and calculates an overall rating for each possible instantiation using a combination of the strength of each preference and the closeness of a potential match. The instantiation with the highest rating will be considered the *best* choice for the user, and, if the best choice is different from and substantially better than the user's proposal, our system will generate natural language utterances to suggest this alternative to the user.

[1]Variables will be represented as a string preceded by an underscore, where the string indicates that variable's type.

Detecting and Modeling User Preferences

Each time an attribute-value preference is deduced, it will be recorded along with a strength-endorsement pair. Thus if there is multiple evidence for a preference, that preference will have several strength-endorsement pairs. The following sections will describe how the strength and the endorsement of a preference are determined.

Preference Strength

Since some preferences are more important to the user than others, a system's ability to assess the strength of a preference is instrumental to its ability to reliably evaluate alternative actions and make useful suggestions to the user. We argue that an analysis of preference strength must take into account both the utterance in which the preference is conveyed and the *conversational circumstances* under which the preference is expressed.

From our analysis of naturally occurring dialogue, we have identified three ways in which a preference can be explicitly conveyed in an utterance. First, the preference can be expressed directly through a statement such as ''I like AI courses.'' Utterances of this type represent the strongest surface expression of a preference. Second, a preference can be conveyed indirectly in the utterance. For example, suppose that the user requests objects that have a particular attribute value, as in the user query ''Which math courses are available?'' The fact that the user has chosen to ask about courses whose content area is math suggests a preference[2], though a more moderate one than if he had directly stated that he likes math courses. Third, the user can convey an uncertain preference as in ''I don't think I like AI courses'' or ''I might like to take an AI course.'' We call this *hedging* the preference and view it as conveying a weaker preference than the other two forms of expression.

We have also found that the *conversational circumstances*, or the situation in which the preference is expressed, contribute to determining the strength of the preference. For example, a preference is very strong if it is given by the user as the reason for rejecting a solution, since the user is effectively saying that potential solutions with this particular characteristic are undesirable. Similarly, a preference included by the user in the initial description of the problem to be solved is likely to be stronger than a preference expressed in response to a system query since in the first case the user was conscious of the preference at the outset and wanted it to be taken into account in formulating a plan. We use the *conversational circumstances* to determine a range of possible strength for the preference. We have identified four different conversational circumstances as follows:

Reject-Solution (Rej-Soln): The user gives the preference as a reason for rejecting a solution. Example: ''I don't want to take CS360. I don't like networks courses.''

[2]The specified attribute value cannot be a constraint in the current plan. For example, if the user's current goal is to satisfy a math requirement, then asking which math courses are available does not convey a preference for taking math courses.

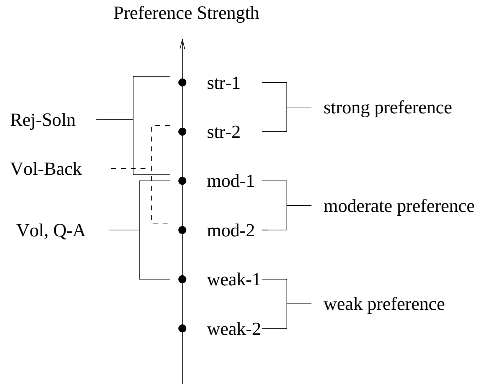

Figure 1: The Possible Range of Preference Strength

Volunteered-Background (Vol-Back): The user includes certain preferences as part of the initial problem description. Example: ‘‘I want to satisfy my seminar course requirement. I’d like to take an AI course.’’

Volunteered (Vol): The user volunteers preference information (without prompting from the system) during the dialogue. Example: (After being informed that CS360 meets at 6pm) ‘‘Great. I like evening courses.’’

Question-and-Answer (Q-A): The user provides preference information in response to a system query. Example: (after being asked what type of courses he is interested in) ‘‘I like networks courses.’’

We maintain six positive preference strengths and six negative preference strengths. Figure 1 illustrates how the conversational circumstances correlate with a range of possible preference strengths in our system[3]. Given the semantic representation of a user utterance, the system analyzes it to determine whether it expresses a preference; if it does, then the way the preference is conveyed determines the strength of the preference within the range established by the conversational circumstance in which the utterance occurs. If the preference is expressed as a direct statement, the strength will be the highest possible within the range. If the preference is conveyed indirectly (as in the utterance ‘‘Which AI courses are available?’’), the strength will be the middle preference strength in the range. If the preference is hedged, the strength will be the lowest in the range. If, for example, the user rejects a solution by saying ‘‘I don’t want to take that course. I don’t like early classes’’ the preference for taking courses that meet early will be *str-1 (neg)*, the highest preference strength within the *Rej-Soln* range.

Preferences can often be identified even though they are not expressed in an utterance. For example, a preference can be deduced by analyzing a pattern of rejections by the user for common features of the rejected solutions. Our system assigns a preference strength of *weak-2* to preferences found in this manner, since the user is a cooperative agent and is expected to explicitly express the preference if it were stronger. In addition, a preference can be attributed to the user on the basis of a user stereotype.

[3] Because positive and negative preference strengths are handled identically, Figure 1 can be viewed as depicting either the positive or negative direction. The direction is denoted by either a (pos) or (neg) following the preference strength.

Reliability Rating	**Endorsements**
Very-Strong	Rej-Soln, Vol-Back
Strong	Vol, Q-A
Moderate	PH-Ded-Str
Weak	PH-Ded-Weak, Stereo
Very-Weak	PH-Ded-Init

Figure 2: Reliability Ratings of Endorsements

Endorsements

Endorsements are explicit records of factors that affect one’s certainty in a hypothesis (Cohen 1985). Cohen introduced the endorsement-based approach to reasoning under uncertainty and endorsements have since been used in belief models (Galliers 1992). We use endorsements to represent the evidence that the system has accumulated to support its belief about a user preference. In our system, this evidence corresponds to the method used in detecting the preference. This allows us to clearly evaluate the quality of the evidence for each preference.

The use of endorsements in the recognition of preferences gives us several advantages. The evidence for a preference can be allowed to accumulate and combine so that several weak endorsements can build up into a stronger endorsement. Also, because the endorsement represents the evidence that the system has for a preference, the system can use the endorsements to explain this evidence to the user if questions about its suggestions arise. The use of endorsements also allows the generation component to explicitly choose what evidence is sufficient for various uses. For example, in proposing a solution to the user at the beginning of a dialogue, the system might take into account user preferences with a lower reliability rating than it would in suggesting a better alternative to the user’s proposal.

Figure 2 contains a list of the endorsements that we use in our system. The endorsements are arranged into classes which reflect the reliability of the evidence represented by each endorsement. These classes are similar to Murray’s endorsement reliability classes (Murray 1991).

Because our endorsements correspond to the methods by which the preferences are detected, each conversational circumstance is represented by an endorsement. The ranking of these endorsements also corresponds to their preference strength ranges. This correlation exists because of the collaborative nature of consultation dialogues. If a preference is strong, then it is important to the consultation process and the user will more clearly express it, thus leading to more reliable evidence. The endorsements which do not match the conversational circumstances (*PH-Ded-Str*, *PH-Ded-Weak*, *Stereo*, and *PH-Ded-Init*), represent methods for detecting preferences which are not explicitly expressed in the dialogue. The following sections describe these methods.

Proposal History As the problem solving process progresses, both the system and the user will propose various actions. These proposals can be made directly as in ‘‘You should take CS360'' or ''I would like to take CS882.'' They can also be made indirectly. For example, relating a user's query such as ''Who is teaching CS360?'' to the existing domain plan might require inferring a chain of domain actions that are not already part of the plan, including actions such as *Take-Course(U, CS360)*. These inferred actions explain *why* the user asked the question and are actions that the user is implicitly proposing be added to the plan.

We maintain a data structure called a *Proposal History* (PH) which keeps track of proposals by recording the status of the proposal (accepted, rejected, current, or suspended), and the information that the system has explicitly given the user about the proposal (such as the instructor and meeting time of a course for a proposed *Take-Course* action). The PH allows the system to record trends in the user's processing of proposed solutions so that it can detect preferences which are not explicitly expressed. If, for instance, the user has rejected several proposals, the objects involved in these proposals may have common factors which could reflect a preference. For example, the user may have rejected several *Take-Course* actions, each of which involved a course that is known to meet early in the morning or to be very difficult. This checking is done not only with the rejected proposals in the PH but also with the accepted proposals.

Processing the PH enables us to capitalize on our use of endorsements. Associated with each proposed action in the PH is the set of attributes of objects involved in the action[4] whose values have been conveyed during the dialogue[5]. When a proposal is accepted or rejected, these characteristics will be recorded as possible preferences with the endorsement *PH-Ded-Init*. This is an extremely weak endorsement since little can be deduced from a single instance. The Proposal-History-Deduction (PH-Ded) endorsements are later combined. If there are two *PH-Ded-Init* endorsements, they are replaced by a *PH-Ded-Weak* endorsement. A third *PH-Ded-Init* endorsement results in a *PH-Ded-Str* endorsement. Thus the number of proposals demonstrating the pattern will determine the endorsement.

Default Stereotypical Information In order to capture preferences which are typical of a class of users, we maintain a set of stereotypical preferences. These preferences can be of any strength, but all will receive the endorsement *Stereo* when attributed to the user. While stereotypical information can be useful, it is not as reliable as information that is detected through actual discourse, so the endorsement is low in the reliability ranking and can be overridden by preferences deduced from the dialogue.

Attaching the Preferences to Actions

We stated earlier that preferences are held with respect to particular goals or actions. An individual might prefer a moderately difficult course when she takes a course for credit, while preferring difficult courses when she audits a course. As the dialogue progresses, our system is working to recognize the user's goals and build an inferred plan. Therefore, if the user is building a plan to take a course and we recognize a preference, we attach it to the *Take-Course* action. When a preference is recognized, it is attached to all of the domain actions in the user's inferred plan that contain the relevant object (in this case, _crs) as a parameter.

[4] These objects are the parameters of the action.

[5] In the future, stereotypical knowledge about attribute values will also be used.

Preference	**Preference Strength**
difficulty(_crs,moderate)	mod-1 (pos)
meets-at(_crs,<10am)	mod-2 (neg)

Figure 3: *Take-Course(_usr,_crs)* Stereotypical Preferences

Preference	**Strength**	**Endorsement**	**Utt**
content(_crs,AI)	mod-1 (pos)	Vol-Back	(2)
prof(_crs,Smith)	mod-1 (pos)	Vol	(8)
difficulty(_crs,moderate)	mod-1 (pos)	Stereo	
	mod-1 (pos)	Vol	(10)
meets-at(_crs,>6pm)	str-1 (neg)	Rej-Soln	(12)
meets-at(_crs,<10am)	mod-2 (neg)	Stereo	
	str-1 (pos)	Rej-Soln	(13)

Figure 4: Preferences Recognized for *Take-Course(U,_crs)*

Example of Preference Detection

The following simple example demonstrates how our preference recognition process works. Suppose that the stereotypical user preferences for the action *Take-Course(_usr,_crs)* are those in Figure 3, and the following dialogue occurs.

(1) U: I need to satisfy my seminar course requirement.
(2) Which AI courses are there?
(3) S: CS882, CS883 and CS889.
(4) U: What is CS882?
(5) S: Natural Language Processing.
(6) U: Who is teaching CS882?
(7) S: Dr. Smith.
(8) U: Great. What is the difficulty level?
(9) S: CS882 has a moderate difficulty level.
(10) U: Good. What time does CS882 meet?
(11) S: 6pm.
(12) U: Then I don't want to take CS882 since I don't like evening courses.
(13) I prefer courses which meet before 10am.

Figure 4 shows the preferences detected for the *Take-Course* action, the strengths and endorsements attached to the preferences, and the utterances which convey the preferences. This example does not illustrate use of the Proposal History since it only comes into play in longer dialogues.

Using Preferences in Response Generation

In a collaborative environment, if the consultant knows of a substantially superior alternative to an action proposed by the user, but does not suggest it to the user, she cannot be said to have fulfilled her responsibility as a collaborative agent. Thus, an intelligent consultation system must use its beliefs about the user's preferences to identify user proposals that it believes are suboptimal and to suggest better alternatives to the user. When a user has proposed an action, either explicitly (e.g., *Can I take CS360?*) or implicitly (e.g., by asking a question such as *Who is teaching CS360?*, from which our plan recognition component (Lambert & Carberry 1991) would infer a *Take-Course* action), our response system (Chu-Carroll & Carberry 1994) will evaluate the specific action proposed by the user and determine whether a substantially better alternative exists. To accomplish this, the system invokes the **ranking advisor**, described in the next section, to rate different instantiations of the proposed action that will accomplish the user's goal. If a substantially better alternative is identified, the system will propose the alternative action and support its proposal with the attribute values that it believes make the alternative preferable.

The Ranking Advisor

The ranking advisor's task is to determine how the parameters of an action can best be instantiated, based on the user's preferences. For each object that can instantiate a parameter of an action (such as CS883 and CS889 for instantiating _crs in *Take-Course(U,_crs)*), the ranking advisor extracts the values of its attributes from the knowledge base and receives the user's preferences relevant to the action from the preference recognition component. A threshold is used to eliminate preferences with low reliability ratings so that only preferences with reliability ratings above a certain level will be allowed to affect the system's decision. For the purpose of suggesting better alternatives, our system uses preferences with reliability ratings of at least *Moderate*.

Two factors should be taken into account by the ranking advisor: the *strength* of a user preference for an attribute value and the *closeness of a match*. The preference strength represents the importance the user attaches to a preference. The closeness of the match (*exact*, *strong*, *weak*, and *none*) measures how well the actual and the preferred values of an attribute match. It is measured based on the *distance* between the two values where the unit of measurement differs depending on the type of the attribute. The closeness of the match must be modeled in order to capture the fact that if the user prefers difficult courses, a moderate course will be considered preferable to an easy one, even though neither of them exactly satisfies the user's preference.

For each candidate instantiation, the ranking advisor records the preference strength for the relevant attributes and computes the closeness of each match. A rating is computed for each candidate instantiation by summing the products of corresponding terms of the preference strength and the closeness of the match. The instantiation with the highest rating is considered the *best* instantiation for the action under consideration. Thus the selection strategy employed by our ranking advisor corresponds to an *additive* model of human decision-making (Reed 1982).

Example of Ranking Two Objects Suppose the previous dialogue is continued as follows:

(14) U: *What is CS883?*

Our plan recognition component would infer *Take-Course (U,CS883)* as a domain action that the user is considering.

Attribute	**CS883**	**CS889**
prof	Brown	Smith
difficulty	moderate	easy
workload	light	moderate
meets-at	2-3:15pm	10:30-11:45am
content	{expert systems, AI}	{planning, AI}

Figure 5: Domain Knowledge for the Ranking Advisor

We demonstrate the ranking advisor by showing how two different instantiations, CS883 and CS889, of the *Take-Course* action are ranked. The ranking advisor obtains user preferences from the preference recognition component[6], and extracts the attribute values for the two instantiations (Figure 5) from the knowledge base.

The ranking advisor matches the user's preferences against the domain knowledge for CS883 and CS889. For each attribute for which the user has indicated a preferred value, the advisor records the preference strength and the closeness of the match for each instantiation. For instance, in considering the attribute *difficulty*, the preference strength will be *mod-1 (pos)*, and the closeness of the match will be *exact* and *strong* for CS883 and CS889, respectively[7]. Table 1 summarizes the preference strength and the closeness of the match for the attribute values for both candidates. Numerical values are then assigned and a final weight is calculated for each candidate instantiation. In this example, the normalized weight for CS883 is $24/54$ and that for CS889 is $44/54$; thus taking CS889 is considered a substantially superior alternative to taking CS883 for this particular user.

Suggesting Better Alternatives

If the ranking advisor finds a substantially superior alternative to the action proposed by the user, it will notify the user, suggest its proposed alternative, and justify this substitution with evidence. The evidence is chosen by examining the information used by the ranking advisor (Table 1) and selecting the attributes that make CS889 substantially better than CS883. In this example, CS889 scores higher than CS883 in the attributes *prof* and *meets-at*; thus the two attributes and their values are used as justification for the suggested alternative. The system then generates the following utterances:

(15) S: *Taking CS889 is a better alternative than taking CS883.*
(16) *The professor of CS889 is Dr. Smith.*
(17) *CS889 meets from 10:30 to 11:45am.*

These utterances initiate a negotiation with the user concerning the desirability of the user's proposed plan. Whereas Cawsey *et al.*'s system (Cawsey *et al.* 1993) initiates negotiation dialogues with the user based on discrepancies between the system's and the user's domain-related beliefs

[6] The evidence for the preferences in Figure 4 was combined based on the reliability of the endorsements. The *Vol* and *Rej-Soln* endorsements for the difficulty(_crs, moderate) and meets-at(_crs, <10am) preferences outweighed the *Stereo* endorsements.

[7] The closeness of match is determined based on the possible values of difficulty: *very-easy*, *easy*, *moderate*, *difficult*, and *very-difficult*.

Preference	Strength		CS883-Match		CS883-Weight	CS889-Match		CS889-Weight
Content(_crs,AI)	mod-1(pos)	4	exact	3	12	exact	3	12
Prof(_crs,Smith)	mod-1 (pos)	4	none	0	0	exact	3	12
Diffculty(_crs,moderate)	mod-1 (pos)	4	exact	3	12	strong	2	8
Meets-at(_crs,>6pm)	str-1 (neg)	-6	none	0	0	none	0	0
Meets-at(_crs,<10am)	str-1 (pos)	6	none	0	0	strong	2	12
					24			44

Table 1: The Preference Strengths and Closeness of Matches

and intentions, our system also does so because of detected better alternatives based on user preferences.

Related Work

Previous work in user modeling has focused on such issues as inferring and utilizing user goals (Carberry 1988; McKeown, Wish, & Matthews 1985; McKeown 1988), tailoring explanations to the user's level of expertise (Paris 1988; Chin 1989), and inferring user knowledge (Kass 1991), as well as other relevant sources of information. However, none of the above take into account user preferences. The systems which utilize user preferences either obtain the information by explicit user model acquisition (Rich 1979; Morik 1989), or assume that the preferences are provided to them (van Beek 1987).

Van Beek's (van Beek 1987) system is capable of analyzing the user's plan and suggesting better alternatives, based on (Joshi, Webber, & Weischedel 1984). Our model improves upon van Beek's in that it 1) dynamically recognizes user preferences, 2) treats user preferences as potential influencers (Bratman 1990) of a plan rather than as goals, and 3) takes into account both the strength of the preference and the closeness of the match in evaluating user proposals.

Conclusion

This paper has addressed the recognition and exploitation of user preferences. We have presented a mechanism for dynamically recognizing user preferences during the course of a collaborative task-oriented dialogue. Our mechanism utilizes both the utterance conveying a preference and the conversational circumstances in which the utterance occurs. In addition, it can detect preferences that are communicated implicitly through a pattern of accepted or rejected proposals. We have also presented a strategy for exploiting the model of user preferences to detect suboptimal solutions and suggest better alternatives. By taking user preferences into account, our system is better able to fulfill its role as a collaborative agent.

References

Bratman, M. 1990. What is intention? In Cohen, P.; Morgan, J.; and Pollack, M., eds., *Intentions in Communication*. MIT Press. chapter 2, 15--31.

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

Cawsey, A.; Galliers, J.; Logan, B.; Reece, S.; and 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 the Simulation of Behaviour*, 130--139.

Chin, D. 1989. KNOME: Modeling What the User Knows in UC. In Kobsa, A., and Wahlster, W., eds., *User Models in Dialog Systems*. Springer-Verlag. 74--107.

Chu-Carroll, J., and Carberry, S. 1994. A plan-based model for response generation in collaborative task-oriented dialogues. In *Proceedings of AAAI*. To Appear.

Cohen, P. 1985. *Heuristic Reasoning about Uncertainty: An Artificial Intelligence Approach*. Pitman Publishing Company.

Galliers, J. 1992. Autonomous belief revision and communication. In Gardensfors., ed., *Belief Revision*. Cambridge University Press.

Joshi, A.; Webber, B.; and Weischedel, R. 1984. Living up to expectations: Computing expert responses. In *Proceedings of AAAI*, 169--175.

Kass, R., and Finin, T. 1988. Modelling the User in Natural Language Systems. *Computational Linguistics* 14(3):5--22.

Kass, R. 1991. Building a User Model Implicitly from a Cooperative Advisory Dialog. *User Modeling and User-Adapted Interaction* 1(3):203--258.

Lambert, L., and Carberry, S. 1991. A tripartite plan-based model of dialogue. In *Proceedings of the ACL*, 47--54.

McKeown, K.; Wish, M.; and Matthews, K. 1985. Tailoring explanations for the user. In *Proceedings of the IJCAI*.

McKeown, K. 1988. Generating Goal-Oriented Explanations. *International Journal of Expert Systems* 1(4):377--395.

Morik, K. 1989. User Models and Conversational Settings: Modeling the User's Wants. In Kobsa, A., and Wahlster, W., eds., *User Models in Dialog Systems*. Springer-Verlag. 364--385.

Murray, W. 1991. An endorsement-based approach to student modelling for planner-controlled tutors. In *Proceedings of IJCAI*, 1100--1106.

Paris, C. 1988. Tailoring Object Descriptions to a User's Level of Expertise. *Computational Linguistics* 14:64--78.

Reed, S. 1982. *Cognition: Theory and Applications*. Brooks/Cole Publishing Company. chapter 14, 337--365.

Rich, E. 1979. User Modelling via Stereotypes. *Cognitive Science* 3(4):329--354.

van Beek, P. 1987. A model for generating better explanations. In *Proceedings of the ACL*, 215--220.