A Model for User-Specific Explanations from Expert Systems

Peter G. van Beek
Department of Computer Science

Research Report CS-86-42
September 1986

A Model for User-Specific Explanations from Expert Systems

by

Peter G. van Beek

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, 1986

Abstract

In this thesis we present a computational model for generating non-misleading, user-specific explanations from expert systems. Ideally an expert system should, as an aid in formulating cooperative responses, both maintain a model of the user from the ongoing dialogue and possess knowledge of a user's expectations of cooperative expert behavior. Our model focuses on how knowledge of the user's goals, plans, and preferences should influence a response. Included are two important specifications: what information about the user is needed plus an algorithm for using that information to compute user-specific responses. The explanation model may be seen as extending the work of Joshi, Webber, and Weischedel to include user-specific goals and to specify the algorithm independent of domain.

The algorithm, together with the model of the user, present a general method of computing the responses enumerated by Joshi et al. They also allow us to generate helpful responses that address a particular user's preferences and goals and to recognize cases where a direct, correct response may violate the user's expectations of cooperative expert behavior and thus mislead or confuse the user. This involves, among other things, the ability to: provide a correct, direct answer to a query; explain the failure of a query; compute better alternatives to a user's plan as expressed in a query; and recognize when a direct response should be modified and make the appropriate modification.

While we focus in this thesis on explanations in the context of expert advice-giving systems, we feel the approach is applicable to a broad range of question types and to expert system explanation generation in general.

Acknowledgements

I am particularly grateful to my supervisor, Robin Cohen, for suggesting the topic of this thesis and for her patient guidance and encouragement during the work. I also wish to thank Marlene Jones and David Poole for their helpful comments on drafts of the thesis. Marlene provided me with many examples of possible explanations from the domain of educational diagnosis for which I offer my thanks. Finally, thanks to family and friends who encouraged me and expressed an interest in my progress. To their oft repeated question, "How's your thesis going?", I can at last give the most satisfactory answer, "It's done!".

Table of Contents

1

Introduction

> Man may smile and smile but he is not an investigating animal. He loves the obvious. He shrinks from explanations. Yet I will go on with mine.
>
> — Joseph Conrad, *The Secret Agent*

Explanation capabilities are an important area for research in computational linguistics. They are of theoretical interest as a sub-problem in the larger area of generation of coherent natural language text. But they are also of practical importance for computer systems in language intensive fields such as computer aided instruction and expert consulting systems. The communication needs of these types of programs have increased the necessity for sophisticated explanation capabilities.

Expert consulting systems have great potential for filling a need in highly specialized problem domains where human experts are scarce or unavailable. Unlike many complex computer systems, such as an operating system, where an ability to explain its actions is often considered superfluous, the ability for an expert system to explain its reasoning and actions can be regarded as essential. Any knowledge based system incapable of explaining its behavior would be unlikely to be accepted by its users or by those human experts in the same field of knowledge. A recent study of 200 physicians confirmed that high-quality explanations were an important requirement for a diagnostic expert system (Buchanan and Shortliffe 1984).

1.1. What is an Explanation

> The positivist separation of logic and pragmatics meant that for many years pragmatics was the Cinderella of language, forced to stay home and do the dirty work while sisters syntax and semantics received all the attention.
>
> — Alan Garfunkel, *Forms of Explanation*

While the meaning of the concept of explanation seems intuitively clear, an attempt to construct an adequate definition of what constitutes an explanation encounters difficulties. Webster's (Merriam-Webster Inc., Springfield, Mass., 1985) defines an explanation as:

> "Something that explains, makes plain or understandable, gives the meaning of, accounts for, or gives an interpretation of"

In the context of expert systems, we wish to define an explanation as something that explains an action, event, fact, or result to a user of the system; makes it plain or intelligible to him, makes him understand it, or justifies it to him.

The difference between these two definitions lies in the relativistic sense of the latter definition. Explanations can now be ordered by how explanatory they are to a particular person. These explanations may be the result of an implicit or explicit question on the part of the user of the system. We will examine in detail the explanations that result from information-seeking requests (as used in Carberry 1983a) where the questioner seeks plan related information but where the plan is not executed within the dialogue. Here the response should include an indication of the possibility or advisability of an action and an answer to the implicit questions of why the result is true and what are the consequences of the action. The response should also suggest better alternative courses of action when the recognized plan of the user is non-optimal.

There is a large body of work on developing a formal syntactic and semantic theory of scientific explanation. A notable and representative example is the covering law or deductive model of explanation developed by Carl Hempel (1965). This model tells us that F is explained by a statement of boundary conditions B and law L if and only if F is deducible from B and L in such a way that B and L are both essential to the deduction. Explanation in the covering law model is not a matter of making something understood by any particular person. Rather, in Hempel's view, pragmatic considerations are the sorts of things a scientific model of explanation would abstract from. But, in the same way that a logically well constructed argument can fail to convince a listener, a logically well constructed explanation can fail to be explanatory.

As our adopted definition of explanation indicates, there is a pragmatic component to explanation: it requires reference to the person to whom the explanation is directed. It is often the case that in explaining the answer to some question, Q, to someone, U, one must provide not only a correct (direct) answer to Q, but also such information as may be required to bring U to understand, accept, and not be mislead by the explanation. In addition to providing a correct (direct) answer to Q, the explanation may have to provide additional background information, correct certain of U's mistaken background beliefs that are inconsistent with the correct answer to Q, show U that certain of his correct background beliefs are in fact consistent with the answer (Matthews 1981; Bromberger 1965), address U's motivation for asking Q (McKeown, Wish, and Matthews 1985), and provide additional information aimed at preventing U from performing inappropriate reasoning (Joshi, Webber, and Weischedel 1984a, 1984b).

Thus, we will argue that an explanation should depend on the beliefs, goals, knowledge, and assumptions of a user on a particular occasion. We characterize what is necessary and sufficient in an explanation such that it is understandable and satisfactory to a particular user and avoids misleading that user. We then demonstrate that the pragmatic processing needed to produce these user-specific explanations requires a model of the user. As a step in this direction we show how a model of the user that incorporates some information on the background, goals, plans and preferences of the user can be used to produce user-specific explanations. The type of domain independent reasoning that the explanation facility must go through to make use of the user model will be detailed. A case will be made that these explanations are indeed better than those that result from the absence of such pragmatic processing. In support of the feasibility of incorporating models of the user into an explanation facility, an existing implementation will be described.

1.2. Major Points of the Proposed Model

This thesis presents a model for generating user-specific explanations, based on the background of the user, the user's goals, plans for achieving the goals, and preferences among the goals. Our work should be seen as extending the research of Joshi, Webber, and Weischedel (1984a, 1984b), hereafter referred to as Joshi's for simplicity. Joshi's work addresses the problem of when a speaker should supply additional information to prevent inappropriate reasoning on the part of the hearer (see chapter 4 for a detailed account) and can be summarized in the following maxim:

> If you, the speaker, plan to say anything which may imply for the hearer something that you believe to be false, then provide further information to block it.

Further, he gives a formal model of a user's beliefs about cooperative expert behavior that can be used to design a system that will avoid misleading responses to task related questions.

Joshi approach is to explicitly list each possible case - a potentially impossible task in more complex domains. We design an algorithm to actually reason about a user's goals and plans. This has the advantage of providing a more general method of computing these non-misleading responses and of providing some domain independence in that the explanation procedures and the procedures for reasoning about the goals, plans and preferences of the user remain constant across domains. With a model of participants in a conversation as purposive agents as motivation, the algorithm is also designed to, with the aid of a model of the user, discover new classes of cases which would require an implemented system to supply additional information to prevent inappropriate reasoning. Finally, by reasoning about a user's goals, plans and preferences among goals, the system can offer alternatives to the user which are specific to that user. The model works by constructing possible user domain plans and comparing them to system-generated plans, to provide the best alternative to a user's query. As a result, the model is particularly useful for detecting and correcting user's misconceptions.

1.3. Outline of the Thesis

The next chapter contains a brief discussion of expert systems, examples of early work in generating explanations from expert systems, and an outline of some of the problems with these previous approaches.

Explanations can be improved in a number of ways beyond addressing the goals and plans of a user, including enhancing the knowledge base of the expert system, optimizing the surface form of the explanation, and taking into consideration the sophistication of the user. A discussion and examples of the research done in these ways of improving explanations can be found in chapter 3. Chapter 3 concludes with a discussion of related work on inferring and responding to the goals of a user of an expert advice-giving system, with an emphasis on our differences.

Truthful and informative responses are, of course, expected from an expert but they are not in themselves sufficient in that they may imply for the hearer conclusions that the expert believes to be false. This is the subject of chapter 4. Chapter 4 also contains a detailed description of the work of Joshi (1984a, 1984b) in preventing false inferences upon which the current research is largely based.

Chapter 5 is the heart of the thesis, presenting both the motivation for the model for generating user-specific explanations and a detailed description of it. The model was implemented in a student-advisor domain and further tested by hand simulating responses to requests for explanations in the context of an expert system for the diagnosis of learning disabilities. Examples from these two domains are presented to further illustrate our ideas about explanations. Chapter 5 also contains the details of the Prolog implementation of the model in the student-advisor domain.

The thesis concludes in chapter 6 with a discussion of the unfinished aspects of the model and possible further extensions, some speculative remarks, and a summary of the contributions and promise of the approach.

2

Explanations and Expert Systems

Archie: *Ah! - I knew there was something! - McFee's dead.*
George: *What?!!*
Archie: *Shot himself this morning, in the park, in a plastic bag.*
George: *My God! Why?*
Archie: *It's hard to say. He was always tidy.*

— Tom Stoppard, *Jumpers*, Act II

An expert system can be characterized as "a computer program that embodies the expertise of one or more experts in some domain and applies this knowledge to make useful inferences for the user of the system" (Hayes-Roth, Waterman, and Lenat 1983). Expert systems, also known as knowledge-based systems or computer consulting systems, can be further distinguished from conventional computer programs by, among other things, the ability to employ self knowledge to provide explanations and justifications. That these explanations and justifications are not, at present, ideal is shown through a summary of the explanation capabilities of three existing expert systems: MYCIN (Buchanan and Shortliffe 1984) and Teiresias (Davis 1982) for bacterial diagnosis and therapy, and XPLAIN (Swartout 1983) for digitalis therapy. A description of other prominent expert systems, the techniques employed in building them, and an extensive bibliography can be found in (Hayes-Roth, Waterman, and Lenat 1983). Subsequent chapters of this thesis will detail our proposals for improving these explanations.

2.1. Description of Rule-Based Expert Systems

A key problem in building expert systems is how to represent the human expert's knowledge in a usable form. Many expert consulting systems, including the three discussed below, employ rule-based deduction. In these systems, the knowledge of human experts in the problem domain is represented as a possibly large set of simple rules. The rules are used to deduce conclusions from facts supplied by the user and to guide the dialogue between the system and the user. A simple example of rule-based deduction will help clarify the subsequent discussion of current research in explanation capabilities.

Facts:	dog(Fido) meows(Myrtle) ¬barks(Fido) human(John)
Rules:	human(x) and smiles(x) => friendly(x) dog(x) and wags-tail(x) => friendly(x) happy(x) => wags-tail(x) happy(x) => smiles(x) friendly(x) and ¬barks(x) => ¬afraid(y,x) likes-milk(x) and meows(x) => cat(x)

Figure 2.1: Simple Knowledge Base

When expert problem solving capabilities are captured in rule-based deduction, the problem solving can be formalized as a theorem-proving problem, where structures called AND/OR trees are useful for depicting the activity of the theorem prover. Consider the simple knowledge base shown in figure 2.1 (adapted from Nilsson 1980). The proposition we wish to prove is whether there exists a cat and a dog such that the cat is not afraid of the dog. The theorem prover seeks a consistent solution tree terminating in fact nodes to prove the proposition. The derivation (AND/OR) tree in figure 2.2 depicts the deductive reasoning.

Some facts necessary to continue the deduction are not present in the knowledge base and the user is asked to supply them. This also provides the user with an opportunity to ask questions. Note that the AND successor nodes in the tree are linked by semi-circles indicating that all of the successor nodes must be solved in order to solve the parent node. The OR successor nodes indicate that only one of the successors must be solved. This is closely related to the problem reduction heuristic with which humans solve problems: turning hard problems into possibly several simpler problems. The reduction approach to problem solving can also be represented by an AND/OR tree. Thus the proof procedure behaves as a problem solver and also implicitly defines a derivation tree.

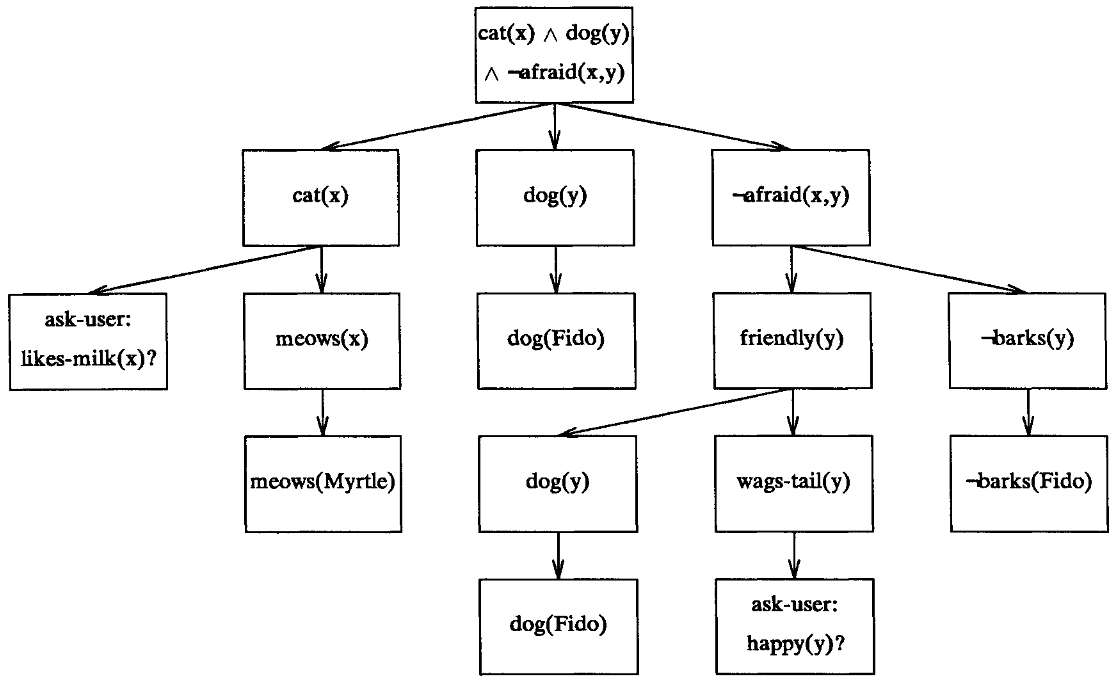

Figure 2.2: Derivation Tree

2.2. Research in Explanations

One approach to explanations is to use stored text to answer a user's queries. This is exemplified in conventional systems or programs by manual pages or help screens. This approach is generally considered inadequate in that it makes it difficult to give explanations at varying levels of detail, to base explanations on previous discourse, and to take into account *why* a user asked the question.

Recent research in providing explanation capabilities for expert systems has concentrated on producing explanations directly from the rules and facts in the knowledge base. To facilitate this, a history is kept of all rules that were applied in the deduction and an indication of whether the rule succeeded or failed. (The history is conceptually very similar to the depiction of the derivation tree given above.) An explanation then involves translating into natural language portions of the derivation tree. This approach has the added advantage of being linguistically relevant. As noted above, humans often perform problem solving by problem reduction and justify their actions and results by explaining the steps in the reduction and how those steps follow, one from another.

Early work towards explanations in MYCIN (Buchanan and Shortliffe 1984), a medical consulting system, allows the user to ask the system "why" a question was being asked. The resulting explanation is a translation of the rule currently under consideration. Subsequent "whys" to the same question causes the translator to move up one level in the derivation tree. Given the rule-based deduction depicted graphically in figure 2.2, the dialogue in figure 2.3 could be a typical question and explanation interaction with MYCIN (user input is shown in bold type, comments are in italics).

As can be seen from the examples, one can see a portion of the rules, but to understand the overall reasoning scheme, a user needs to request a display of each rule that was used. How the rules fit together is never made explicit by the system, in part because such knowledge is often lacking in the system. Rules are often designed for efficiency and omit the underlying causal

Is Fido happy?

Why?

{ *User wants to know why the system is asking whether Fido is happy.* }

If Fido is happy then he wags his tail.

Is Fido happy?

Why?

{ *User asks why again to the same question. The explanation given is a result of moving up one level in the derivation tree and translating the rule.* }

If Fido wags his tail and is a dog then he is friendly.

Figure 2.3: Hypothetical MYCIN Example (1)

mechanisms that explain why the consequent of a rule follows logically from its antecedents. For example, the empirical or causal relationship between being happy and smiling (see figure 2.1) is not included in the knowledge base. Thus a question regarding this relationship could not be answered except to quote the rule.

More recent work in the MYCIN domain (Buchanan and Shortliffe 1984) has attempted to rectify the limitation of explaining one rule at a time. The system has the ability to answer questions regarding chains of reasoning. To do this, a measure of complexity is associated with rules and concepts . The user specifies his level of expertise before requesting an explanation. Reasoning chains that involve simpler concepts as intermediates are collapsed to avoid the display of information that might be obvious to the user. As well, concepts that are gauged as too complex for the user are also omitted (see figure 2.4). Only those concepts whose complexity falls within the range defined by the expertise level and the level of detail will be mentioned in the explanation. Broader inference statements are generated to bridge the concepts that are chosen.

This approach preserves the logical flow of the explanation without introducing concepts of inappropriate complexity. Rules are omitted from the explanation if they are either too simple or too complex, depending on the user's level of expertise. Again with respect to the previously depicted derivation tree and assuming the appropriate complexity and importance measures attached to the rules, the following could be an exchange in this system:

Expertise 3

{ *User specifies expertise* }

Is Fido happy?

Why?

{ *Why is the system asking whether Fido is happy? The rule: happy(x) => wags-tail(x) is omitted from the explanation because it involves a simple concept the user can be presumed to know because of his specified expertise level. The rule : dog(x) and wags-tail(x) => friendly(x) is omitted because its complexity measure places it above the range of expertise specified by the user.* }

Happiness is associated with friendliness.

Not barking and friendliness means not afraid.

Figure 2.4: Hypothetical MYCIN Example (2)

While MYCIN allows the user to examine the reasoning and underlying knowledge at various levels of detail, there are serious shortcomings to this approach to explanation. To foreshorten the possibility of repetition, these shortcomings will be detailed after a short discussion of the Teiresias expert system (Davis 1982), since they apply equally well to that case.

Teiresias builds on the MYCIN work. As with previous work, Davis takes as a starting point the assumption that the program's approach to its domain is sufficiently intuitive that a recap of the deduction is a reasonable basis for an explanation. He asserts that, since the rules are formulated by human experts and embody accepted patterns of human reasoning, they have adequate explanatory value (Davis 1982). Davis further argues that backward chaining and modus ponens are an accepted and often used (though probably implicitly) problem solving method and therefore there exists some framework for viewing the program's actions that allow a recap to be comprehensible and acceptable to the user.

Possible questions to Teiresias include how and why. The user can control the detail with a numeric argument (see figure 2.5). The rules have static importance based on their CF (Certainty Factor) value. Definitional rules (CF = 1.0) have no information, while those with smaller CF's have progressively more information. The entire "distance" from the current node to the top of the goal tree is normalized to 10, and the argument following the WHY or HOW is taken as some fraction of that distance. The chosen chain is then compressed into a single answer, the compression being based on the CF values of the rules involved. Figure 2.5 contrasts the difference in the amount of detail the explanations provide depending on the numeric argument to the

In light of the site from which the culture was obtained and the method of collection, do you feel that a significant number of ORGANISM-1 was obtained?

** WHY

[i.e., Why is it important to determine whether ORGANISM-1 was observed in significant numbers?]

[1.0] This will aid in determining whether there is therapeutically significant disease associated with this occurrence of ORGANISM-1.

It has already been established that

[1.1] the method of collection is sterile,

therefore, if:

[1.2] the organism was observed in significant numbers, and the site of the culture is not one of those which are normally sterile, then there is strongly suggestive evidence (0.9) that there is therapeutically significant disease associated with this occurrence of the organism.

Also: there is strongly suggestive evidence (0.8) that the organism is not a contaminant.

** WHY 4

{ The "distance" to the top at this point is eight. The argument of four to the WHY command indicates the user wants four-tenths of the present "distance" to be summarized. Since the three steps in the chain constitute roughly four-tenths of the "distance" to the top, they are compressed into a single answer. }

We are trying to find out whether the organism has been observed in significant numbers, in order to determine an organism or class of organisms for which therapy should cover.

Figure 2.5: Teiresias Example

"Why" command. The explanation resulting from "Why" with no numeric argument defaults to the level of the rule currently under consideration, whereas the one resulting from "Why 4" summarizes four-tenths of the current reasoning chain into a single answer.

In Teiresias, as in MYCIN, different levels of detail can be presented to a user based on a scheme of assigning static complexity measures to the rules in the knowledge base. Unfortunately, in this scheme, two rules with the same complexity measure are assumed to have the same information content. But a user who is familiar with only the first of them will find much more information in the second. In both MYCIN and Teiresias, "why" is underspecified and must necessarily be expanded by the system to avoid misinterpretation by the user (Davis 1982). For example, "why" might mean "why are you asking about that test instead of ...". Neither system can determine the reason why a user requested an explanation. Both MYCIN's and Teiresias' model of explanation relies solely on a trace or recap of the program's actions for generating explanations. But an explanation may have to correct a user's misconceptions, reaffirm what a user knows, or provide information aimed at preventing inappropriate reasoning (recall our discussion in the introductory chapter of what an explanation should address) to be truly explanatory.

An alternative approach to having the user specify a level of expertise or detail, and the one that this thesis ascribes to, is to have the system infer from the user/system dialogue and the system's augmented knowledge of the domain, not only the level of expertise of the user, but also his specific goals in order to generate user-specific explanations. This theme will be taken up again beginning in chapter 3.

We conclude this chapter with an account of the XPLAIN expert system (Swartout 1983). As mentioned previously in the discussion on MYCIN, many expert systems are limited to describing what they did (or the reasoning path they followed to reach a conclusion), but they are incapable of justifying their actions (or reasoning paths), where justification here means the reasoning behind the system (Swartout 1983). In short, previous systems can state what they did but cannot state why they did it. To remedy this deficiency, Swartout uses automatic programming to include knowledge of causal relationships and classification hierarchies into an expert system. A translation of the rules along with this additional information provides much better explanations. Swartout applied his ideas to an existing expert system. Figure 2.6 contrasts *before* and *after*: an explanation from the original expert system and an explanation from the XPLAIN expert system which has been augmented with knowledge of causal relationships and classification hierarchies. A choice of two viewpoints (computer or medical) are allowed to be attached to steps to indicate to whom the step should be explained. This allows some customization of explanation based on what the user is interested in. As well, this system can produce shortened explanations if several of the causal relationships have previously been explained (see latter part of figure 2.6).

The XPLAIN system does indeed generate improved explanations, but it too does not address the goals of the user, the intent of the question, or any misconceptions the user may hold. Thus, it suffers from many of the same shortcomings as the previous two example systems.

Before:

What is the level of serum potassium? **Why?**

{ *User questions why he is being asked to supply the value of serum potassium.* }

My top goal is to begin therapy. One step in doing that is to check sensitivities. I am now trying to check sensitivity due to potassium.

After:

Please enter the value of serum potassium: **Why?**

The system is anticipating digitalis toxicity. Decreased serum potassium causes increased automaticity, which may cause a change to ventricular fibrillation. Increased digitalis also causes increased automaticity. Thus, if the system observes decreased serum potassium, it reduces the dose of digitalis due to decreased serum potassium.

Please enter the value of serum potassium: **3.7**

Please enter the value of serum calcium: **Why?**

{ *The system produces a shortened explanation, reflecting the fact that is has already explained several of the causal relationships in the previous explanation. Also, since the system remembers that it has already told the user about serum potassium, it suggests the analogy between the two here.* }

The system is anticipating digitalis toxicity. Increased serum calcium also causes increased automaticity. Thus, (as with decreased potassium) if the system observes increased serum calcium, it reduces the dose of digitalis due to increased serum calcium.

Please enter the value of serum calcium: **9.0**

Figure 2.6: Example from Swartout's Digitalis Therapy Advisor

3

Better Explanations

With a sweeping brush, problems in natural language generation can be distinguished as deciding *what to say* and *how to say it*. In some research these two aspects are tightly coupled, in other research the solution to one is assumed in order to focus on the other. In Teiresias (see chapter 2), for example, choosing what to say is decided by taking the derivation tree as a basis and "pruning" it according to a user-inputted level of expertise. The rules that are left are then simply translated into their English equivalent. Recent research has investigated how to better decide *what to say* and *how to say it*. We detail below some representative examples of this research into better explanations or responses. In particular, the first part of this chapter will discuss: improving surface choice (McKeown and Derr 1984), the use of rhetorical predicates for deciding what to say and how to organize it (McKeown 1982; but see also Mann and Thompson 1983), and, in a vein similar to rhetorical predicates, the use of different descriptive strategies for naive and expert users (Paris 1985). "The unifying theme of much current pragmatics and discourse research is that the coherence of dialogue is to be found in the interaction of the conversants' *plans*. That is, a speaker is regarded as planning his utterances to achieve his goals ... [and] the hearer attempts to infer the speaker's goal(s) and to understand how the utterance furthers them" (Cohen and Levesque 1985). This chapter concludes with a discussion of how the goals and plans of a user can be inferred from the user/system dialogue and used towards generating improved user-specific explanations as exemplified in the work of Allen (1983); Carberry (1983b); McKeown, Wish, and Matthews (1985); and Pollack (1984).

3.1. Surface Generation

Given a semantic representation of what to say, a surface generator must construct an appropriate surface structure. For example, an expert system may select part of the goal tree, a particular goal and its antecedent subgoals, to explain a behavior of the system. This is passed to the surface generator which translates the logical propositions into English. One decision to be made in translating is when to use a sequence of simple sentences or a single complex one.

McKeown and Derr (1984) have looked at how focus can be used in deciding when to use a sequence of simple sentences or a single complex one, in selecting sentence voice (active or passive), and in determining whether to replace a noun with a pronoun. See figure 3.1 for an example of using temporary focus shift in forming complex sentences using subordination. The actual semantic representation is horn clause logic and the input to the surface generator is a portion of the goal tree selected to explain a behavior of the system. The figure shows an English translation for simplicity sake.

The generator determines that subordination is necessary by checking whether focus shifts over a sequence of three propositions. Each proposition has five arguments: verb, protagonist, goal, beneficiary, and focus.

Rule: If the first proposition focuses on FOCUS1 and mentions an unfocused argument GOAL1, and if the second proposition specifies GOAL1 as its focus, but in the third proposition the focus returns to FOCUS1, then the first and second propositions can be combined using subordination.

Before: Assembly language has a prerequisite.
{ *FOCUS1 = Assembly language, GOAL1 = prerequisite* }
The prerequisite was taken.
{ *focus = GOAL1* }
Assembly language does not conflict.
{ *focus = FOCUS1* }

After: Assembly language has a prerequisite that was taken.
Assembly language does not conflict.

Figure 3.1: Determining whether subordination is necessary using focus

3.2. Rhetorical Predicates

In contrast to the supposition underlying the Teiresias and MYCIN explanation facilities, namely that a trace of the rules used in the deduction is sufficiently intuitive and explanatory, McKeown (1982) states that communicative strategies that people are familiar with should be used to effectively convey information. She examines texts and classes each sentence as one of a set of rhetorical predicates. Some combinations of predicates are considered more likely than others. These standard combinations are encoded as schemas which represent the structural relation between propositions in a text. As well, some schemas are considered more appropriate than others for certain discourse situations. For example, in situations where an object or concept can be described in terms of its sub-parts or sub-classes, the "constituency" schema (figure 3.2) is used. The "compare and contrast" schema (not shown) is used in response to a question about the difference between objects. To use the schemas in response generation, the input question is used to produce a relevant knowledge pool. Depending on the information available to answer the question, a particular schema is chosen for formulating the response. In figure 3.2, the "constituency" schema is selected for describing a telephone receiver because the relevant knowledge pool contains much information about sub-parts and their properties. A formal representation of the answer is constructed by selecting propositions from the knowledge pool which match the rhetorical techniques in the given schema. For clarity reasons, figure 3.2 shows only the English equivalent.

3.3. Descriptive Strategies for Different Users

Explanations are most effective if they are based on the expected level of knowledge of the recipient of the explanation. For example, an explanation on how a car engine works would probably be fundamentally different if for a music student than if for a student of mechanical engineering. Typically this doesn't consist of merely giving more details for the novice as is often thought (Davis 1982; and see chapter 2). Paris (1985) characterized the explanation strategies employed for naive and expert users in describing complex physical objects by examining junior

Notation: "{}" indicate optionality, "/" indicates alternatives, "+" indicates the item may appear 1-n times, and "*" indicates that the item may appear 0-n times.

{ Constituency }
Cause-effect* / Attributive*
  { Depth-identification / Depth-attribute
    { Particular Illustration / Evidence }
    { Comparison; Analogy } }+
  { Amplification / Explanation / Attributive / Analogy }

1) The receiver includes a ring-shaped magnet system around a coil and a ring shaped armature of anadium Permendur. 2) Current in the coil makes the armature vibrate in the air gap. 3) An attached phenolic-impregnated fabric diaphragm, shaped like a dome, 4) vibrates and sets the air in the canal of the ear in motion.

Depth-attributive for the receiver

1. Depth-Attributive
2. Cause-effect
3. Depth-Attributive
4. Cause-effect

Figure 3.2: Constituency Schema Example

and adult encyclopedia entries. She found that in the adult entries, the details given are mainly about the sub-parts and their properties and only very briefly about the mechanical processes involved. This corresponded to the constituency schema of McKeown (figure 3.2). In the junior entries, the process mechanism is more important than the sub-parts and given in more detail. More specifically, the junior entries contained a process description of the object, the cause and effect relations, and functional information about the objects parts. As well, there were no large gaps in the chain of inference. Paris found that there was no corresponding schema or other organizing structure for juniors. The main strategy in describing objects to a naive user is to trace through the process that allows the object to perform its function.

3.4. User-Specific Explanations

Previous research has investigated user modeling in the context of interpreting an utterance correctly and of generating appropriate explanations. In MYCIN and Teiresias (see chapter 2), the user can specify his level of expertise when requesting an explanation. In Paris' research (see section 3.3), different descriptive strategies are employed depending on whether the user is an expert or naive user. While these approaches allow the explanation to be tailored to the general level of knowledge of the user, they do not address the specific knowledge of the user, they require the system to have a priori knowledge of the user's expertise, and, more importantly in our context, they do not address *why* an explanation was requested nor do they recognize that a direct response may be misleading.

An alternative approach is to gather information about the user's goals, assumptions, and knowledge from the user/system dialogue (e.g. Allen 1983; Carberry 1983b; McKeown, Wish, and Matthews 1985; Pollack 1984). This approach is particularly fruitful in the expert system environment where the user and the system work cooperatively towards common goals through the dialogue and the user's utterances may be viewed as actions in plans for achieving those goals. The present work draws on this latter approach. In particular, our model uses knowledge of the plans and goals of the user in order to produce user-specific, cooperative responses. Actually deriving these plans and goals is not part of our work. But below we describe some closely related work which ideally could be used to gather the goals needed in our explanation algorithm.

3.4.1. Recognizing Intentions from Natural Language Utterances

In their early and influential work on plan recognition, Perrault, Allen and Cohen (1978) observe that "part of cooperative behavior is the detection by one agent of obstacles in the plans he believes the other agent holds, possibly followed by an attempt to overcome them". Allen (1983) gives as example the following exchange between a patron and a clerk at an information booth in a train station:

Patron: When does the Montreal train leave?
Clerk: 3:15 at gate 7.

In this exchange, the clerk not only answers the patron's direct question, but also provides as additional information the departure gate.

To model the generation of responses that provide more information than strictly required, Allen adopts a view of language use as goal-oriented behavior. This view has its basis in the speech act theory of language use, developed most notably by Austin (1962) and Searle (1969), which can be summarized in Searle's suggestion that "a theory of language is part of a theory of action". Allen (1983; but see also Allen and Perrault 1980) uses a theory of speech acts based on plans, plan construction, and plan recognition where speech acts are modeled as actions in a planning system.

To return to the patron and clerk example (above), in Allen's model an explanation of what occurs would be as follows: The patron has a goal of boarding the train to Montreal but lacks some information needed in order to achieve the goal. To overcome this obstacle, the patron formulates and executes a plan to question the clerk. For his part, the clerk attempts to infer the patron's goals and plans from the patron's question. In this example, it is straightforward for the clerk to recognize that a goal of the patron is to know when the train departs (Allen's model is also able to handle examples where this step is more problematical because the question is a sentence fragment or is phrased indirectly). But the clerk also infers that the patron wishes to board the train and, being helpful, recognizes that another obstacles exists in achieving this goal (the patron may also need to know the departure gate) and generates a response that overcomes both obstacles.

This plan recognition or inference is done in Allen's system with the use of rules of the form: "If agent S believes agent A has a goal X, then agent S may infer that agent A has a goal Y" in conjunction with knowledge of the expected goals in a particular domain and the use of rating heuristics which judge the likeliness that an inferred goal is part of the the actual intention of the user. Pollack (1984) examines the plan inference rules that may be needed when the user has not accurately and literally expressed a request for advice. She proposes a set of alternate goal rules that combine some of Allen's rules and sketches a preliminary version of an algorithm.

Our work differs from Allen's, and is probably closer in spirit to Pollack's, in that we study different forms of cooperative behavior, such as: non-misleading responses and offering alternatives when the recognized plan of the user is non-optimal or will fail in its intention. As well, we propose tracking the user's goals throughout a dialogue as an aid in producing subsequent responses.

3.4.2. Tracking the User's Goals

In Allen's work, the system is able to infer the complete plan of the user from a single utterance. This is possible because the domain has a small number of possible domain goals. Carberry (1983a) argues that in more complex domains this may not be possible since the user's overall plan may consist of many subplans and subgoals and the user's current goal within such a plan may change during the course of a dialogue. She states that "a cooperative information-provider will assimilate the preceding dialogue, infer the underlying task-related plan motivating the speaker's queries, and focus on that aspect of the task on which the information-seeker's attention is focused." Carberry's approach is to build the complete plan (context model) as the dialogue progresses by inferring a lower-level goal from an utterance and relating it to potential higher level plans. To do this, she distinguishes between the immediate goal, derived goal, and focused goal of a user (local plan context) and the user's overall plan (global plan context or context model). An enriched set of inference rules are used to recognize the goals in the local plan context and focusing heuristics are used (such as "the user will generally obtain all desired information about the currently focused task and the most recently considered subaction before considering other tasks") to relate the focused goal to the global plan and thus to update the context model.

Carberry is primarily concerned with using the context model as an aid to understanding a user's utterances; in related papers (1983b, 1985), she demonstrates in particular how the context model can be used to correct pragmatically ill-formed queries and to understand intersentential ellipses. Our model proposes tracking the user as an aid in generating responses.

3.4.3. Varying the Explanation by Point of View

One way in which the content of an explanation can vary is according to the perspective or point of view taken on the underlying problem domain (McKeown 1984; McKeown, Wish, and Matthews 1985). For example, in the student advisor domain, where users of the advisor typically work towards a degree, there are a number of points of view the student can adopt for selecting courses. The selection of courses can be viewed as a process of meeting requirements (i.e., "how do courses tie in with requirement sequencing?"), as a state model process (i.e., "what should be completed at each state in the process?"), as a semester scheduling process (i.e., "how can courses fit into schedule slots?"), or as a process of maximizing personal interests (as in "how will courses help me learn more about AI?"). Given these different points of view, alternative explanations of the same piece of advice (i.e., yes) can be generated in response to the question: "Should I take both discrete math and data structures this semester?" The different responses could be the following (McKeown 1984):

1. **Requirements**: Yes, data structures is a requirement for all later Computer Science courses and discrete math is a co-requisite for data structures.

2. **State Model**: Yes, you usually take them both first semester, sophomore year.

3. **Semester Scheduling**: Yes, they're offered next semester, but not in the spring and you need to get them out of the way as soon as possible.

4. **Personal Interests (e.g. AI)**: Yes, if you take data structures this semester, you can take Introduction to AI next semester, and you must take discrete math at the same times as data structures.

In more recent research, McKeown (1985) describes an approach to inferring the goals of a user within a brief discourse segment. These inferred goals, or points of view, of the user are then used to vary the response depending on the strength of the system's belief in the inference (see figure 3.3). Specifically, McKeown develops techniques for organizing the knowledge base to support different explanations, for identifying possible goals underlying the current discourse, and for determining when an underlying goal should be taken into account in forming the explanation. Consider dialogue 1 of figure 3.3. The system has a strong belief that the underlying goal of the discourse is taking NLP (from the fact that both questions a and b inquire about taking NLP and because the user stated his future intention in question a) and makes use of this goal in response c to "construct an explanation from the proper perspective" (McKeown, Wish, and Matthews 1985). In dialogue 2 of figure 3.3, the strength of the system's belief in its discernment of the underlying goal of the discourse is not sufficient to warrant taking an underlying goal into account in forming the explanation.

While McKeown's work importantly addresses *why* the user asked a question, i.e. the underlying goal of the question, it does not identify possible user misconceptions, avoid possibly misleading responses, or compare and suggest possible better alternative plans.

1a. S: I'm planning on taking NLP in the future. What are the prerequisites?
plausible goal = take NLP
A: AI.

b. S: Is it offered next semester?
plausible goal = take NLP
relevant goal = take NLP
A: Yes, on Mondays and Wednesdays at 1:10.

c. S: I'm also thinking of taking computability this semester. Would that be a good idea?
plausible goal = take computability
A: Yes, computability is particularly important for NLP since it covers grammars so it's a good idea to take it first.

2a. S: Is natural language offered this semester?
plausible goal = take NLP
A: Yes, at 1:10 on Mondays and Wednesdays.

b. S: I'm thinking of taking computability this semester. Would that be a good idea?
plausible goal = take computability
A: Yes, it's your last requirement and it's a good idea to get it out of the way before going on to electives.

Figure 3.3: McKeown Example

4

Preventing False Inferences

Whether man or machine, it is expected and necessary that an expert respond truthfully and informatively to questions. However, truthful and informative responses are not always sufficient in that they may imply for the hearer conclusions that the expert believes to be false. The expert must recognize these cases and modify or add to the response to block the listener from inferring false conclusions.

Joshi (1984a) proposes a modification of Grice's Maxim of Quality (1975) to which a cooperative expert must conform in responding to questions:

> Grice's *Maxim of Quality* -
> Do not say what you believe to be false or for which you lack adequate evidence.
>
> Joshi's *Revised Maxim of Quality* -
> If you, the speaker, plan to say anything which may imply for the hearer something that you believe to be false, then provide further information to block it.

Several classes of false conclusions that should be blocked by modifying otherwise true responses have been identified in the literature. They include but are not limited to the following:

1. A discrepancy exists between the system's beliefs and those of its user. The user draws false conclusions from what is said because the presuppositions of the response are not in accord with the system's beliefs (Kaplan 1983; McCoy 1984; but see also Mercer and Rosenberg

1984).

2. The user draws false conclusions by using standard default reasoning to conclude (incorrectly) that there is nothing special about this case (Joshi 1984b).

3. The user draws false conclusions in a task-oriented context because of violated expectations about the way a cooperative expert will respond (Joshi 1984a).

Our current work is a continuation of this research into predicting false conclusions and modifying responses accordingly with the emphasis in the order that the above points are described, the majority being on the last point. Thus it will be helpful to explicate these three points in somewhat more detail.

4.1. Belief Discrepancies

In questions directed to an expert, a user may reveal a belief or assumption that is incompatible with the beliefs held by the expert. Kaplan (1983) investigates how to detect and respond to these discrepancies. Consider the following exchange:

> **User:** How many French students failed cse121 last term?
> **System:** I don't know of any French students.

The user shows through the query that he believes, among other things, that the set of French students is non-empty, that there is a course cse121, and that it was given last term (i.e. the presuppositions of the question). Now suppose the system doesn't know about any French students. The technically truthful response of "None", or equivalently, "No French students failed cse121 last term" may mislead the user into believing the false conclusion that the expert concurs with the user's beliefs about French students and further, that the French students did well in the course. Kaplan's system, as a first step in responding to a belief discrepancy, denies the false information.

McCoy (1984) describes a method to infer and correct a user's object-related misconception. In the following example, the user has confused two types of ship:

User: Give me the hull number of all destroyers whose height is above 190.
System: All destroyers that I know about have a mast height between 85 and 90. Were you thinking of the aircraft carriers?

Before continuing the dialogue, the user's incorrect belief must first be addressed. To do this, McCoy classifies object-related disparities according to the knowledge base feature involved. In the exchange above, for example, the user has an attribute misconception and is confusing the object discussed with another object that has the specified property. The response involves denying the wrong information and mentioning the object which does have the property involved in the misconception.

4.2. Inappropriate Default Reasoning

Default reasoning as discussed by Reiter (1980) is a logical system that supports reasoning and drawing conclusions from partial information. Informally, most standard default rules are of the form of "given no reason to suspect otherwise, there's nothing special about the current case". For example, knowing only that Tweety is a bird an agent could conclude by default that since most birds fly so can Tweety fly. Joshi (1984b) examines cases in an interaction with an expert system where the user reasons from the partial information in the response to draw a false conclusion based on inappropriately applying a default rule. The following example (Joshi 1984b) may make this clear:

User: Is Sam an associate Professor?
System: Yes, but he doesn't have tenure.

If the system only responded with "Yes", the user could apply a default rule that says "most associate professors are tenured" to conclude that "Sam is also tenured". The system, therefore, must recognize cases where the user may perform inappropriate default reasoning and modify the response to block it.

4.3. Violated Expectations

Joshi (1984a, 1984b) focuses on characterizing the types of informing behavior usually expected of an expert in responding to questions concerning the achievement of some goal. A general statement of the problem is as follows:

> With respect to an expert's responses to questions, if *Q* expects that *R* would inform him of *P* if *P* were true, then *Q* may interpret *R*'s silence regarding *P* as implying *P* is not true. Thus if *R* knows *P* to be true, his silence may lead to *Q* being mislead (Joshi 1984a).

Thus, an expert system must modify its planned response if there is reason to believe that the response may mislead the user. To identify these cases Joshi makes use of the user's stated goal (the goal directly achieved by using the information requested) and his intended goal (the goal underlying the stated goal of a request).

Scenario:

The stated goal of the user is not being in the course and the intended goal of the user is to avoid failing the course.

User:

Can I drop numerical analysis?

System:

a) Yes, however you will still fail the course since your mark will be recorded as withdrawal while failing.

b) Yes, but a better way is to take an incomplete to have more time to perform the work.

Figure 4.1 Example from Joshi, Webber, and Weischedel (1984a)

In addition to the direct, correct answer, a user expects to be informed if, for example, the stated goal does not achieve the intended goal or if there is a better way to achieve the intended goal (refer to Figure 4.1). If the expert system does not supply this additional information, *P*, it risks misleading the user. The user may interpret the expert system's silence regarding *P* as implying that *P* is not true. In figure 4.1, the user asks "Can I drop course numerical analysis?". Consider response (a). The system determines that it is possible to drop the course since the deadline has not passed. Unfortunately, the result will be a failure on the student's permanent record. If the system gives the straightforward, truthful response of "Yes" this may mislead the user; he may assume the stated answer is also an answer to his intended goal of not failing the course. Thus, the system must add to the direct response to the query to explain that the user's stated goal will fail in its intent. In response (b), the system determines a better plan to achieve the goal of not failing the course and includes this additional information in the response.

In all, Joshi has determined five cases where information may have to be provided by the expert, *R*, in order not to mislead the user, *Q*. These cases are summarized in figure 4.2 and rely on the stated and intended goals shown at the top of figure. As a final example, let us examine response (a.1) more closely. Suppose that it is past the deadline for dropping courses. Thus *R* responds with "You can't drop 577". But if *R* knows another action that would achieve *Q*'s intended goal, *Q* would expect to be informed about it. If not so informed, *Q* may mistakenly conclude that there is no other way to achieve his intended goal. To avoid misleading *Q*, the response includes the additional information: "you can change to audit status". The analyses for the remaining modified responses are similar.

Query: Can I drop course 577?
Intended goal: Avoid failing the course.
Stated goal: Drop the course.

a. Failure of enabling conditions.
   a.1 A way (alternative).
      "You can't drop 577; you can change to audit status."
   a.2 No way.
      "You can't drop 577, the deadline has passed."

b. A non-productive act (consequences).
      "You can drop 577. However, you will still fail."

c. A better way.
      "You can drop 577, but taking an incomplete to have more time to perform the work is a better way."

d. The only way.
      "You can drop 577. No other change of status is possible."

e. Something turning up.
   e.1 External event.
      "You can't drop 577. If 'e' occurs, you won't fail 577."
   e.2 Event followed by action.
      "You can't drop 577. If 'e' occurs, you could 'a' and thus avoid failing."

Figure 4.2: Joshi Summary

Since our model for user-specific explanations has as its starting point Joshi's work on avoiding misleading responses, it will be useful to recount some of the limitations of Joshi's work and to sketch how our model removes these limitations (full details of our model can be found in the next chapter).

Joshi's approach can provide an additional or modified response in order to block inappropriate reasoning on the part of the user because of violations of the user's expectations of cooperative expert behavior. These modified responses also have intrinsic worth in that they provide additional useful information to the user. However, his approach computes the above only in regards to the stated and intended goal of the immediate question, which, as we shall argue, can be inadequate for avoiding misleading responses.

The following example demonstrates how Joshi's modified responses may themselves be misleading: Imagine that a student wishes to learn more about AI and thus asks the system whether he can enroll in a particular course. If the answer is no, perhaps because the student lacks the prerequisites, Joshi's modified response would also inform the student of any alternative AI courses that the student would be permitted to take. Clearly, if the possible alternatives did not help the student in the pursuit of, say, his degree, this additional information would be of little use to the student and providing it may seriously mislead him. Thus Joshi's modified response may itself mislead.

The important point in this example is that the system can not plan to avoid misleading the user with its response by reasoning solely about the stated and intended goals of the query. The system must also have recourse to the user's domain goals and plans. In this case a model of the user that contains the domain goal and plan of the user's academic program would be sufficient. As well, by adding the user's higher domain goals/plans to the user model we can identify additional cases where a response must be modified in order to prevent false inferencing.

As the examples thus far have shown, a major claim of Joshi is that a system, so that it will not produce misleading responses, must recognize when a user's plan is sub-optimal and provide a better alternative. How to generate alternatives and determine "betterness" are left unspecified. Our introduction of the user's higher domain goals (e.g in the course domain: the goal of getting a degree with certain course requirements) and the tracking of the user's goals throughout the dialogue provide some insights into generating and comparing alternatives with regards to a specific user.

To conclude our comparison, Joshi's approach to computing modified responses is to explicitly enumerate all potential cases (see figure 4.3) - thus placing the burden on the author who must prepare for all contingencies; whereas our approach is to reason about the user's goals according to general principles, independently of domain.

```
if admissible(drop(Q,C)(Sc))
   then if ~holds(~fail(Q,C),drop(Q,C)(Sc))
      then begin nonproductive act
         if (E b)[admissible(b(Sc)) & holds(~fail(Q,C),b(Sc))]
            then a way
            else no way
         end
      else if (E b)[admissible(b(Sc)) & holds(~fail(Q,C),b(Sc)) & better(b,f)]
         then a better way
   else if (E b)[admissible(b(Sc)) & holds(~fail(Q,C),b(Sc))]
      then a way
      else no way
   ...
```

Notation:

Q	the user
C	the course
Sc	the current state of the student
admissible(e(S))	event/action e can apply in state S
holds(P,S)	P, a proposition, is true in S

Figure 4.3: Joshi's approach

5

A Model For Better Explanations

Make plans by seeking advice ...

— *Proverbs 20:18*

As previous discussion has indicated, a model of explanation that relies solely on a trace or recap of program actions for formulating explanations may not give the most complete or best explanation, and may even fail to be explanatory. This ineffectiveness is the result of ignoring epistemic (what the user does and does not know), motivational (why the user requested the explanation), and contextual (the user's overall domain goals and preferences) considerations. For an explanation to be explanatory it must provide for an individual user, U, not only a correct, direct answer to a question, Q, but also provide such other information as may be necessary in order to bring U to understand the answer to Q. Thus, in addition to providing a correct, direct answer to Q, an explanation may have to

(i) **provide additional background information** since U may not possess the requisite concepts and foundations (Swartout 1983, chapter 2; Paris 1985, chapter 3). U possibly can remember no reasons to suggest an explanation or lacks specific knowledge.

(ii) **correct certain of the U's mistaken background beliefs** that are inconsistent with the correct answer to Q. If the explanation lacks this the user may be mislead into inferring false conclusions (Kaplan 1983, chapter 4).

(iii) **address U's motivation for asking Q.** In the context of advice-seeking questions, the motivation for asking Q may be the attainment of another, higher goal. One facet includes assuring U that his goals were taken into consideration in arriving at the response (McKeown, Wish, and Matthews 1985, chapter 3).

(iv) **provide additional information** aimed at preventing U from drawing false conclusions from inappropriate default reasoning or violated expectations of how an expert would respond (Joshi 1984a, 1984b, chapter 4).

(v) **and address U's overall goals, objectives, and preferences** among those goals and objectives.

Addressing these pragmatic components of explanation requires a model of the user's beliefs, knowledge, goals, and plans.

In what follows we provide a partial framework in which expert systems could practice the pragmatic philosophy of explanation expressed in points (iv) and (v). We show how a model of the user which contains some general facts about the user along with the objectives, plans, and preferences of the user can be incorporated into the explanation facility of an expert system in furtherance of our goal of good explanations. We demonstrate how the model of the user is constructed and maintained by the system and an algorithm is presented which produces these explanations by reasoning about the goals and plans contained in the model of the user. Our work should be seen as an extension of the approach of Joshi (1984a, 1984b, chapter 4) which characterizes the type of informing behavior usually expected of an expert and states that an expert system must modify its planned response if there is reason to believe that the response may mislead the user.

The algorithm, together with the model of the user, present a general method of computing the responses enumerated by Joshi (see figure 4.4). This involves, among other things, the ability to provide a correct, direct answer to a query; explain the failure of a query; compute better alternatives to a user's plan as expressed in a query; and recognize when a direct response should be modified and make the appropriate modification. The algorithm and the user model thus constitute a processing model for generating user-specific explanations. This model allows us to generate helpful responses that address a particular user's preferences and goals and also to recognize additional cases where a direct, correct response violate the user's expectations of cooperative expert behavior and thus mislead or confuse the user.

As test cases for our model of explanation the application areas of requests for information from a student advisor system and an educational diagnosis expert system were chosen and will be illustrated in this paper. The requests will be about the possibility (e.g., "Can I drop numerical analysis?") or advisability (e.g., "Is the McLeod Phonics Test appropriate?") of an action. An appropriate response will include an indication of the possibility and advisability of the action, better alternatives and consequences of the actions if appropriate, and an answer to the implicit question of why the result should be believed by the user. The model was actually implemented in the student-advisor domain, and further tested by hand simulating responses in the educational diagnosis domain.

The rest of this chapter will proceed as follows. The next section is a brief overview and presents some examples to illustrate what we are proposing. Subsequent to that are separate sections detailing the user model and the algorithm. Toward the end of the chapter are some examples from the student-advisor domain of how the algorithm and the model of the user work together to produce the responses. Finally, the chapter ends with some thoughts on generating responses when the system perceives a misconception on the part of the user. The next chapter concludes the thesis with a discussion of the possible future directions for the model and a summary of the contributions and promise of the approach.

5.1. Overview

Previous work has emphasized that a cooperative expert is expected to infer the immediate goals and plans of an utterance (or discourse segment) and formulate a response in light of these goals and plans. We wish to show that an advice seeker may also expect the expert to respond in light of, not only the immediate goals and plans of the user, but also in light of (i) previously expressed goals or preferences, (ii) goals that may be inferred or known from the user's background, and (iii) domain goals the user may be expected to hold. If the expert's response does not take into consideration these latter type of goals the result may mislead or confuse the user and, at the least, will not be cooperative. To formulate cooperative response, a model of the user is needed to track these three types of goals of the user and also to record some of the background of the user pertinent to the domain. We also wish to demonstrate how the responses enumerated in (Joshi 1984a, see figure 4.4) can be effectively computed given the model of the user.

User:

Can I enroll in CS 375 (Numerical Analysis)?

System:

a) Yes.

b) Yes, but CS 375 does involve a lot of FORTRAN programming. You may find Eng 353 (Technical Writing) and CS 327 (AI) to be useful courses.

Figure 5.1: Example from student advisor domain

For example, consider the exchange between a student and student-advisor system shown in figure 5.1. The user hopes to enroll in a particular course in order to help fulfill his elective requirements. But imagine that in the past the student has told the advisor that he has strong feelings about not using FORTRAN as a programming language. If the student-advisor gives response (a) of "Yes" and the student subsequently enrolls in the course and finds out that it involves heavy doses of FORTRAN programming, the student will probably have justifiably bad feelings about the student-advisor. A better response (b) takes into account what is known about the user's preferences. Thus the system must check if the user's plan as expressed in his query is compatible with previously expressed goals of the user. The system can be additionally cooperative by offering alternatives that are compatible with the user's preferences and also help towards the user's intended goal of choosing an elective (response (b)).

User:

Can I drop CS 273?

System:

a) Yes.

b) Yes, but CS 273 is a co-requisite of CS 265 and you are enrolled in CS 265.

Figure 5.2: Example from student advisor domain

In the example shown in figure 5.2, the user is doing poorly in a course and wants to drop out of it in order to avoid a failing mark on his permanent record. But a plan which leads to the attainment of one goal may result in another goal becoming impossible to achieve. A user may expect to be informed of such consequences, particularly if the goal which cannot now be attained is a goal the user values highly. Knowing the background of the user, the expert can reasonably be expected to know that the student has the goal of getting a particular type of degree. The better response (b), informs the user of the consequences of his proposed plan as concerns the goal of getting his degree.

Finally, avoiding misleading statements is not the only motive for individual models of the users. In responding to questions, experts are expected to be fully informative, they are expected to remind or inform the questioner of the consequences of a course of action. As an example, suppose that a student asks, "Can I drop 225 (technical writing)?" For a user who has already met his Arts credit minimum, the answer could be a simple: "You can drop 225." For another user who could suffer from consequences a more informative response might be something like: "You can drop 225. However, you will still need 3 units of Arts courses in order to get your degree." Here we are not avoiding misleading responses, but rather providing additional useful information. By incorporating a model of the user, explanations can be tailored to specific audiences.

5.2. The User Model

Our model requires a database of domain dependent plans and goals. A plan may contain subgoals, actions, and constraints (cf. Sacerdoti 1977; Litman and Allen 1984). The plans are hierarchical since subgoals may themselves have associated plans. We assume that the goals of the user in the immediate discourse are available by methods such as specified in (Allen 1983; Carberry 1983a; Pollack 1984; see section 3.4.1).

It is proposed that the model of a user contain, in addition to the user's immediate discourse goals, his background, higher domain goals, and plans specifying how the higher domain goals will be accomplished. The plans may be neither completely specified nor instantiated, either because the user has not completely defined his plan or because the system has an incomplete knowledge of that definition. To give this some concreteness, consider an example from the student advisor domain. Initially, the user model will contain some default goals which the user can be expected to hold, such as avoiding failing marks on his permanent record. It will also contain those goals of the user which can be inferred or known from the system's knowledge of the user's background, such as the attainment of a degree. Initially, the plan to obtain a degree will be an incomplete, uninstantiated plan that contains only the constraints imposed by the student's university. Parts of the plan may be corrected (eg. the student changes his major) or expanded and instantiated (eg. the student completes and enrolls in courses) based on the actions of the student. New goals and plans will be added to the model (eg. the student's preferences or intentions) as they are derived from the discourse. For example, if the user displays or mentions a predilection for numerical analysis courses this would be installed in the user model as a goal to be achieved.

5.3. The Algorithm

> The possible is preferable to the impossible.
>
> That is preferable which is the more applicable on every occasion or on most occasions.
>
> — Aristotle, *Topics*, Book III

As our starting point we make the assumption that a user of an expert system can be effectively characterized as a purposive, rational or reasonable agent. Participants in conversation (and, more generally, any human interaction) attribute goals or intentions to their partners in conversation. But the only way in which they are able to employ these goals to effectively understand and respond in a dialogue is by holding fixed the assumption of rationality on the part of the other participants. This assumption of rationality is common to recent research, but is often left implicit. Examples of this recent work are (i) Cohen's (1983, 1984, 1985) model of argument understanding where the speaker assumes the hearer can discern the intentions of the speaker in order to reconstruct the logical structure of the argument and thus achieve understanding. The hearer can comprehend without believing an argument by reconstructing the form of the argument using world knowledge and a model of the speaker's aims, goals, and purposes as they relate to the present context. (ii) Allen's (1983, see chapter 3) work in recognizing intentions from natural language utterances which makes the assumption that "people are rational agents who are capable of forming and executing plans to achieve their goals".

Further, explanations and predictions of people's choices in everyday life are often founded on the assumption of human rationality. The definition of rationality has been much debated, but there is general agreement that rational choices should satisfy some elementary requirements of consistency and coherence. We posit some guiding principles of action that a reasonable agent uses in deciding between competing goals and methods for achieving those goals. Explicitly stating a fuller definition of reasonableness will subsequently be shown to facilitate the generation of user-specific explanations in the context of expert systems.

The principles of action to which a reasonable person will try to adhere are the following (Points 3 and 4 are largely based on Nielson 1979; but see also Cohen and Levesque 1985):

1. An agent will adopt as an immediate goal only those goals in which he believes there exists a method or plan of action that will achieve the goal. An agent may desire a goal that is unachievable but will not execute actions in a plan to achieve the goal if, as far as he can ascertain, some steps in the plan will fail.

2. A plan for achieving a goal will be adopted only if the agent believes that execution of the plan will cause the goal to obtain.

3. If an agent has several compatible goals, he should take the means which will, as far as he can ascertain, enable him to realize the greatest number of his goals.

4. Those goals which are valued absolutely higher than other goals, are the goals which are to be achieved. An agent should seek plans of action which will satisfy those goals, and plans to satisfy his other ends should be adopted only in so far as they are compatible with the satisfaction of those goals he values most highly.

These principles can be viewed as "shared knowledge" between the user and the expert. If the user does not "live up" to these principles, the expert's response should include how the principles are violated and also some alternatives that are better (if they exist) because they do not violate the principles. Because the system is assumed by the user to be a cooperative expert, not informing the user could mislead or confuse the him. Alternatively, the principles can be viewed as a specification of what the program should do.

The algorithm begins by checking whether the user's query is possible or not possible. It then infers the intended goals of the user's query (refer to figure 5.3). If the query is not possible, the user is informed and the explanation includes the reasons for the failure (step 1.0 of algorithm). Alternative plans that are possible and help achieve the user's intended goal are searched for and presented to the user. But before presenting any alternative, the algorithm, to not mislead the user, ensures that the alternative is compatible with the higher domain goals of the user (step 1.1).

If the query is possible, control passes to step 2.0, where the next step is to determine whether the stated goal does, as the user believes, help achieve the intended goal (step 2.1). Previous research has either made the assumption that the intended goal of a query could be uniquely determined from an utterance or kept track of plausible goals until further utterances determined the unique goal. If the system has to answer a question before the goal was uniquely determined, it would have to ask the user for clarification of his goals. Consider the situation where the system believes the user intends to enroll in a course to both achieve his goal of taking numerical analysis courses and to help achieve the goal of getting a degree. The goals are not inconsistent with each other and it is entirely plausible that the user's intent was to take a step in his plan for getting his degree and at the same time satisfying his goal of taking as many NA courses as he can. Of course, if the set of goals is inconsistent, the system must wait for further utterances or ask the user for clarification (Allen and Perrault 1980). (The current

Check if original query is possible.
Infer intended goals.

(1.0) **Case 1:** { *Original query fails* }
    Message: No, [query] is not possible because ...
(1.1) If (∃ alternatives that help achieve the intended goals and
        are compatible with the higher domain goals) then
        Message: However, you can [alternatives]
(1.2) Else
        Message: No alternatives

(2.0) **Case 2:** { *Original query succeeds* }
    Message: Yes, [query] is possible.
(2.1) If not (intended goals) then
        Message: Warn user that intended goal does not hold and explain why.
        If (∃ alternatives that do help achieve the intended goals and
            are also compatible with the higher domain goals) then
            Message: However, you can [alternatives]
        Else
            Message: No alternatives
(2.2) Else If (stated goal of query is incompatible with the higher
        domain goals) then
        Message: Warn user of incompatibility.
        If (∃ alternatives that are compatible with the higher domain
            goals and also help achieve the intended goals) then
            Message: However, you can [alternatives]
        Else
            Message: No alternatives
(2.3) Else If (∃ alternatives that also meet intended goals but are
        better than the stated goal of the query) then
        Message: There is a better way ...
    Else
        { No action }

Figure 5.3: Explanation Algorithm

implementation of our model does not check for the inconsistency of goals, but we feel this is important for a future improved implementation.) Given that the user presents a plan that he believes will accomplish his intended goals, the system must check if the plan succeeds in its intentions (step 2.1 of algorithm).

---

1. The i-goal may already be true.
2. The i-goal may not be possible to achieve.
3. Several relationships that can hold between a stated goal (s-goal) and an intended goal (i-goal) can be enumerated (Joshi 1984a):
    3.1 The s-goal may be the same as the i-goal.
    3.2 The s-goal may be a subgoal that addresses only part of the i-goal. For example, the user's s-goal may be to enroll in a natural language course while his i-goal may be to concentrate on AI.
    3.3 The s-goal may be a pre-condition of the i-goal. For example, the s-goal may be to get read/write access to a file, while his i-goal may be to alter it.
    3.4 The i-goal may be more specific than the s-goal. For example, the s-goal may be to know how to send files to someone on another machine, while his i-goal is just to send a file to a particular machine, which may allow for a specialized procedure.

Figure 5.4: Relationship among goals

To recapitulate, the user model contains the stated goal (s-goal) and the intended goal (i-goal) upon which the current discourse is focused, and some higher domain goals. A representative list of things which must be checked for to identify possible misconceptions on the part of the user and thus to provide the best response, are shown in figure 5.4. The user may falsely believe that a certain relationship holds between his goals; it is then up to the system to correct the misconception. Given our knowledge representation in terms of goals and plans for achieving those goals, a natural method of computing whether these relationships hold presents itself. Thus, (3.1) corresponds to checking for equality, (3.2) involves checking whether the s-goal could be a possible step in the plan for the i-goal, (3.3) whether the s-goal is a pre-condition in the plan for the i-goal, and (3.4) whether the i-goal is among the (possibly) multiple plans available for achieving the s-goal.

As is shown in the algorithm, if the relationship does not hold or the plan is not executable, the user should be informed. Here it is possible to provide additional unrequested information necessary to achieve the goal (cf. Allen 1983). If possible, the system should also inform the user of (1) any pre-conditions that can't possibly become true, given the state of the world, and thus preclude the possibility of achieving the intended goal, and (2) whether the usefulness of the stated goal is dependent on the result of first achieving an all together different goal.

We have identified several ways in which an explanation can be user-specific by providing additional relevant information that is aimed at blocking inappropriate reasoning or at cooperative. In planning a response, the system should ensure that the current goals, as expressed in the user's queries, are compatible with the user's higher domain goals (step 2.2 in algorithm). For example, a plan which leads to the attainment of one goal may lead to the non-attainment of another when a previously formed plan becomes invalid or may make the sub-goal impossible to achieve. A user may expect to be informed of such consequences, particularly if the goal which cannot now be attained is a goal the user values highly (see figure 5.2).

The expert system can be additionally cooperative by suggesting better alternatives if they exist (step 2.3 in algorithm). Furthermore, both the definitions of better and possible alternatives are relative to a particular user.

5.4. An Example

Until now we have discussed a model for generating better, user-specific explanations. A test version of this model has been implemented, in Waterloo UNIX Prolog. Below we present an example to illustrate how the algorithm and the model of the user work together to produce these responses and to illustrate some of the details of the implementation.

Given a query by the user, the system determines whether the stated goal of the query is possible or not possible and whether the stated goal will help achieve the intended goal. In the hypothetical situation shown in figure 5.5, the stated goal of enrolling in cs572 is possible and the intended goal of taking a numerical analysis course is satisfied[1]. The system then considers the

[1] Recall that we are assuming the stated and intended goals are supplied to our model. This particular intended goal, hypothetically inferred from the stated goal and previous discourse, was chosen to illustrate the use of the stated, intended, and domain goals in forming a best response. For a more standard intended goal, like "satisfy the 500 level requirement", the conflict between stated and intended goal would be handled in a similar fashion to the conflict between stated and domain goal, shown in this example.

background of the user (in the course domain: the courses taken), the background of the domain (in the course domain: what courses are offered) and a query from the user (e.g. "Can I enroll in cs572?"), and ensures that the goal of the query is compatible with the attainment of the overall domain goal.

Scenario:

The user asks about enrolling in a 500 level course. Only a certain number of 500 level courses can be credited towards a degree and the user has already taken that number of 500 level courses.

Stated goal: Enroll in the course.
Intended goal: Take a numerical analysis course.
Domain goal: Get a degree.

User:

Can I enroll in CS 572 (Linear Algebra)?

System:

a) Yes.

b) Yes, but it will not get you further towards your degree since you have already met your 500 level requirement. Some useful courses would be CS 673 (Linear Programming) and CS 674 (Approximation).

Figure 5.5: Example from student advisor domain

In this example, the user's stated goal of enrolling in a particular course is incompatible with the user's higher domain goal of achieving a degree because several pre-conditions fail. That is, given the background of the user the goal of the query to enroll in cs572 will not help achieve the domain goal. Knowledge of the incompatibility and the failed pre-conditions are used to form the first sentence of the system's response.

To suggest better alternatives, the system goes into a planning stage. There is stored in the system a general plan for accomplishing the higher domain goal of the user. This plan is necessarily incomplete and is used by the system to track the user by instantiating the plan according to the user's particular case. The system considers alternative plans to achieve the user's intended goal that are compatible with the domain goal. For this particular example, the system discovers other courses that the user can add which will help achieve the higher goal.

To actually generate alternatives and to check whether the user's stated goal is compatible with the user's domain goal, a module of the implemented system is a Horn clause theorem prover, built on top of Waterloo Unix Prolog, with the feature that it records a history of the deduction. The theorem prover generates possible alternative plans by performing deduction on the goal at the level of the user's query. That is, the goal is "proven" given the "actions" (e.g. enroll in a course) and the "constraints" (e.g. prerequisites of the course were taken) of the domain. In the example of figure 5.5, the expert system has the following Horn clauses in its knowledge base:

course (cs673, numerical)

```
get_degree(Student, Action) <-
      receive_credit(Student, Course, Action);
get_degree(Student, []);

receive_credit (Student, Course, Action) <-
      counts_for_credit (Student, Course),
      enrolled (Student, Course, credit, Action),
      do_work (Student, Course),
      passing_grade (Student, Course);
receive_credit (Student, Course, Action) <-
      enrolled (Student, Course, credit, []),
      enrolled (Student, Course, incomplete, Action),
      complete_work (Student, Course),
      passing_grade (Student, Course);

counts_for_credit (Student, Course) <-
      is_500_level (Course),
      500_level_taken (Student, N), lt (N, 2);
counts_for_credit (Student, Course) <-
      is_600_level (Course),
      600_level_taken (Student, N), lt (N, 5);
```

Figure 5.6: Simplified domain plan for course domain.

course (cs674, numerical)

Figure 5.6 shows a portion of the simplified domain plan for getting a degree. For example, a course will count for credit if it is a 500 level course and fewer than two 500 level course have already been counted for credit (since in our hypothetical world, at most two 500 level courses can be counted for credit towards a degree). Similarly for a 600 level course.

The domain plan is then employed to generate an appropriate response. The clauses can be used in two ways: (i) to return an action that will help achieve a goal and (ii) to check whether a particular action is a possible step in a plan to achieve a goal. In the first use, the *Action* parameter is uninstantiated (a variable), the theorem prover is applied to the clause, and, as a result, the *Action* parameter is instantiated with an action the user could perform towards achieving his goal. In the second case, the *Action* parameter is bound to a particular action and then the theorem prover is applied. If the proof succeeds, the particular action is valid step in a plan; if the proof fails, it is not valid and the history of the deduction will indicate why. In this example, enrolling in cs673 is one of the valid steps or actions in a plan for achieving a degree.

Note that the system will generate alternative plans even if the user's query is a step in the domain plan. In this case the challenge is to find a better solution for the user. The (possibly) multiple plans are then potential candidates for presenting to the user. These candidates are then pruned by ranking them according to the heuristic of "which plan would get the user further towards his goals". Thus, the better alternatives are the ones that help satisfy multiple goals or multiple subgoals. One way in which the system can reduce alternatives is to employ previously derived goals of the user such as those that indicate certain preferences or interests. In the course domain, for instance, the user may prefer taking numerical analysis courses. For the example in figure 5.6, the suggested alternatives of cs673 and cs674 help towards the user's goal of getting a degree and the user's goal of taking numerical analysis courses (a middle digit of seven refers to numerical analysis) and so are preferable[2].

[2] Note that in this example the user's intended goal also indicates a preference. Other user preferences may have been previously specificed; these would be used to influence the response in a similar fashion.

The system has the ability to further reduce the alternatives displayed to the user. This can be done by employing domain dependent knowledge. For the course domain a rule of the form: "A mandatory course is preferable to a non-mandatory course", may help eliminate presentation of certain options.

In addition, the system could combine these two methods. Consider the educational diagnosis domain where a user asks the system about the learning disabilities of a child. A response may indicate which further tests the child should be given. Determining these tests involves reference to the case history of the child being diagnosed (model of child) and domain knowledge on what makes a test better. A rule such as, "Tests in which the mode of presentation or the mode of response is the same as that in which the child is having difficulty are less appropriate than those in which they differ" could be used. But the system could also consider the user's preferences in determining which tests should be administered by suggesting tests with which the user is most familiar.

5.5. Joshi Revisited

The discussion in the previous section showed how our model can recognize when a user's plan is incompatible with his domain goals and present better alternative plans that are user-specific. Here we present examples of how our model can generate the responses enumerated by Joshi (see figure 4.2). The examples illustrate different aspects of our algorithm and also demonstrate how our model removes some of the limitations of Joshi's approach (see section 4.3).

```
?consult(example_1);

?       query(change_status(andrew, 577, credit, nil),
              not_fail(andrew, 577, Action));

No, change_status(andrew, 577, credit, nil) is not possible because ...
      fail:  less_than(date, drop/add-deadline)
There are no alternatives.

yes

?consult(example_2);

?       query(change_status(andrew, 577, credit, nil),
              not_fail(andrew, 577, Action));

Yes, change_status(andrew, 577, credit, nil) is possible.
But, not_fail(andrew, 577, _461) is not achieved since ...
      fail:  not_presently_failing(andrew, 577)
However, you can ...
      alts:  change_status(andrew, 577, credit, incomplete)
This will also help towards receive_credit(_10, _11, _461)

yes

?consult(example_3);

?       query(change_status(andrew, 577, credit, nil),
              not_fail(andrew, 577, Action));

Yes, change_status(andrew, 577, credit, nil) is possible.
But, there is a better way ...
      better: change_status(andrew, 577, credit, incomplete)
Because this will also help towards receive_credit(_10, _11, _627)

yes
```

Figure 5.7: Sample responses

Figure 5.7 shows three different responses to the same question: "Can I drop CS 577?" The student asking the question is doing poorly in the course and wishes to drop it in order to avoid failing it. The goals of the query are passed to the Prolog implementation and the response generated depends on these goals, the information in the model of the user, and on external conditions such as deadlines for changing status in a course. For example purposes, the domain information is read in from a file (e.g. consult(example_1)). Figure 5.8 shows the clausal representation of the domain goals and plans used in this example. Consider the first clause of the *receive_credit* predicate. This clause states that the student will receive credit for a course if the course can be counted for credit, the student is enrolled in the course, does the work, and gets a passing grade.

Example 1: In this example, the student's stated goal (dropping the course) is not possible (see (a) of figure 4.2). This is case 1.2 of the algorithm. To formulate the response, the theorem prover is applied to the stated goal and the resulting history of the deduction indicates the reasons why the goal is not possible. These reasons are then presented to the student.

Example 2: In this example, the stated goal is possible, but it fails in its intention (dropping the course doesn't enable the student to avoid failing the course; see (b) of figure 4.2). This is case 2.1 of the algorithm. The system now looks for alternatives that will help achieve the student's intended goal and determines that two alternative plans are possible: the student could either change to audit status or take an incomplete in the course. The plan to take an incomplete is presented to the user because it is considered the best of the two alternatives; it will allow the

```
get_degree(S,Action) <-
      receive_credit(S, C, Action);
get_degree(S,[]);

receive_credit (S, C, Action) <-
      counts_for_credit (S, C), enrolled (S, C, credit, Action),
      do_work (S, C), passing_grade (S, C);
receive_credit (S, C, Action) <-
      enrolled (S, C, credit, []), enrolled (S, C, incomplete, Action),
      complete_work (S, C), passing_grade (S, C);

not_fail (S, C, Action) <-
      not_presently_failing (S, C), enrolled (S, C, credit, []),
      enrolled (S, C, nil, Action);
not_fail (S, C, Action) <-
      enrolled (S, C, credit, []), enrolled (S, C, incomplete, Action),
      finish_work (S, C);
not_fail (S, C, Action) <-
      enrolled (S, C, credit, []), enrolled (S, C, audit, Action);

enrolled (S, C, After, change_status (S, C, Before, After)) <-
      change_status (S, C, Before, After);

change_status (S, C, Before, After) <-
      student (S), course (C, Area),
      enrolled (S, C, Before, []), possible (Before, After);

possible (credit, audit) <- less_than (date, audit-deadline);
possible (credit, nil) <- less_than (date, drop/add-deadline);
possible (credit, incomplete) <- less_than (date, incomplete-deadline);
```

Figure 5.8: Not failing example

student to still achieve another of his goals: receiving credit for the course.

Example 3: Here the query is possible (the student can drop the course) and is successful in its intention (dropping the course does enable the student to avoid failing the course; see (c) of figure 4.2). The system now looks for a better alternative to the student's plan of dropping the course (case 2.3 of algorithm) and determines an alternative that achieves the intended goal of not failing the course but also achieves another of the student's domain goals: receiving credit for the course. This better alternative is then presented to the student.

5.6. Misconceptions

In our discussion we have noted that a technically correct answer to a question can potentially mislead the listener. This was found to be the case when there exists a discrepancy between the beliefs of the user and those of the expert (cf. Kaplan, 1983; see chapter 4). We extend this idea to the beliefs the user may hold about the relationships between his goals and plans for achieving those goals and the beliefs the expert has about those goals and plans. This allows us to identify and present to the user knowledge that may make the misconception clear.

As a simple example, suppose a student believes there is a course called cs377 and asks the system, "Can I enroll in cs377?" If no such course exists, a direct response of "No" would mislead the student. Our system would respond with the equivalent of "No such course exists". This much is straightforward since the system was designed to report the reasons for the failure of a query (Kaplan's work involved databases where this ability was not part of the initial design).

Scenario:

The child appears to have a language difficulty. He doesn't speak much and when he does his sentences are short and choppy with connecting words, prepositions, and articles usually missing. Testing is aimed at getting a measure of intelligence to determine whether this disability is a reflection of mental retardation.

User:

Is the Neale Analysis of Reading an appropriate test?

System:

a) The Neale is a useful test of reading comprehension and reading within context, but it is not an I.Q. test. The PPVT (Peabody Picture Vocabulary Test) would be useful in this situation.

b) While the Neale may prove to be a useful test later in the diagnosis process, if it is felt necessary to assess the child's reading skills, the general level of performance indicates that testing should first be aimed at getting a measure of intelligence to determine whether this disability reflects mental retardation or a learning disability. The PPVT is a good test to give in this situation.

Figure 5.9: Example from educational diagnosis domain

A more complex example is shown in figure 5.9. Here the explanation varies according to the intended goal a user holds when a question is asked and thus the misconception recognized. In both of the cases in figure 5.9, the explanation points out that a goal of the user is inappropriate, provides background information aimed at relieving the user of his misconception, and suggests a better alternative. Response (a), formulated at step 2.1 of the algorithm, informs the user that the stated goal (administering the Neale test) does not achieve his intended goal (administering an I.Q. test). Response (b), formulated at step 2.2 of the algorithm, addresses not the relationship between the stated goal (administering the Neale test) and its intended goal (assessing reading skills), but instead addresses the inappropriateness of the stated goal as it relates to the user's higher domain goal of planning a remedial program for the child under consideration. The system determines that the usefulness of the stated goal is dependent on the result of first achieving an all together different goal. In this case, the user's goals are necessary for computing a correct and best response to the user's queries. An explanation that did not address the misconception of the user may be unsatisfactory to the user and may even fail to be explanatory. The present implementation is not able to detect that the user is attempting to execute a step in a plan out of its proper order which is necessary for response (b).

The example just presented illustrated how the explanation can vary according to the goals of a user and two possible responses to the same question where shown, with the appropriate response depending on the user's intended goal. In general, however, it is very difficult to infer different goals from the same surface structure. Augmenting the user model to include the beliefs of the user may help but a system may have to rely on the user being more explicit about his goals through the dialogue (see section 6.2).

6

Future Work and Conclusion

In this chapter we recount some of the limitations of the current work, describe possible future avenues of research for extending the model we presented for generating better explanations and, lastly, present our conclusions.

6.1. Some Current Limitations

The model of the user and the algorithm for utilizing the model presented in the last chapter are concerned primarily with the user's background and goals. Some concerns remain with the present implementation and the representation used.

- The ability to determine a better plan to achieve the user's intended goal may suffer from large search spaces in less restricted domains. Rather than look at all the user's goals during this process, the search should be more focused, perhaps by grouping the goals according to what actions they accept.

- In the present implementation the representation of plans and the planning process is too simplistic. In particular, the frame problem is not adequately dealt with and there are restricted notions of developing and executing plans over time. As a consequence, the reasoning about the effects of plans isn't as elegant as is desirable.

This has been acceptable so far in our restricted domain, where the emphasis has been on demonstrating the usefulness of such notions in generating explanations, but further work needs to be done. As is well known, planning is difficult and a planning component may not always be able to compose a plan to achieve a goal.

As has been previously noted, our work assumes the existence of methods for inferring the user's goals from his utterance. This is not an unrealistic assumption as much work has been done in this area (see Chapter 3). However, in cases where the user does not explicitly state his goals, these goals are only hypothesized and are not known with certainty. A neglected aspect of modeling a user in our work is the correction of the model when further information indicates that an hypothesis was erroneous. As a complicating factor, some goals, such as the goal of finding a good diagnosis in the educational diagnosis domain, should not be defeasible.

Finally, the future should include work on a component for mapping between the Horn clause representation used by the program and the English surface form. Kindersley (1986) investigates a parser for the educational diagnosis domain.

6.2. Modeling the User's Knowledge About the Domain

We have argued that a system for producing explanations should track the user's goals, plans, and preferences throughout the dialogue and make use of them in formulating a response. Modeling the user's knowledge about the domain would also be useful in several ways for tailoring a response to a specific user.

One such way is in trimming a response. To explain something to a person, it is often sufficient to call his attention to some particular fact of which he has not properly taken recognition. It may be possible, based on some beliefs about the user's knowledge of the domain, to trim a response to be in harmony with Grice's Maxim of Quantity: "be only as informative as is required for the current purposes of the exchange" (1975).

In some domains it is desirable for an expert system to support explanations for users with widely diverse backgrounds. To provide this support an expert system should tailor the content of its explanations according to the user's knowledge of the domain. An expert system currently being developed for the diagnosis of a child's learning disabilities and the recommendation of a remedial program provides a good example (Jones and Poole 1985; McLeod and Jones 1985). (Some of our examples are drawn from this domain and were provided by Marlene Jones.) Psychologists, administrators, teachers, and parents are all potential audiences for explanations. As well, members within each of these groups will have varying levels of expertise in educational diagnosis. Cohen and Jones (1986) illustrate the need for different responses through the following example:

User:

Could the student's mispronunciation "errors" be due to dialect?

Response to Parent:

Yes, non-standard pronunciations may be due to dialect rather than poor decoding skills.

Response to Psychologist:

Yes, the student's background indicates the possibility of a dialectal difference.

The parents know that their child speaks a dialect. What they lack and wish to know is some specific domain knowledge. For the psychologist, the reverse is true; the domain knowledge is known but the particulars of the child are unknown. Cohen and Jones (1986) provide a framework for varying the content of an explanation based on incorporating a model of the user. The user model begins with default assumptions based on the user's group (psychologist, teacher, or parent). The model is then updated as new information becomes available as the dialogue progresses. In formulating a response, the system determines which information is relevant to answering the query and includes that portion of the information which is believed to not be part of the shared knowledge between the user and the system.

Finally, a model of the user's beliefs would also be useful in correcting a user's misconceptions. In the Neale example (figure 5.9), several possibilities exist for the user's misconception, such as (i) user is misinformed about case data or is confused about the purpose of a test (facts), (ii) or user draws wrong conclusion from a test or case data (deduction). The best explanation is one which addresses directly the user's underlying misconception.

6.3. Additional Question Types

Several additional question types seem to be amenable to the solution we propose for questions as to the possibility or advisability of a task. For example, explanation seeking questions of the standard type "Why is it the case that *p*?" are often prompted by the belief that another alternative, q, should have been the case. And in this event, the questioner may not feel satisfied if he is offered an explanation of why p is the case (as in the MYCIN examples of chapter 2).

System:

Administer an individual intelligence test. For this particular case, the Peabody Picture Vocabulary Test (PPVT) is recommended.

User:

Why is this test being recommended?

System:

The PPVT is being recommended because it does not require verbal responses and the student appears to have language problems including difficulties with expressive language. Hence, the PPVT is more appropriate than more common tests such as the WISC-R or Stanford Binet.

Figure 6.1 Example from Cohen and Jones (1986)

In the example shown in figure 6.1 (due to Cohen and Jones, 1986), the user questions the system's recommendation of a particular intelligence test, preferring instead to administer a more familiar intelligence test, such as the Stanford Binet. Notice that the user is not asking "why administer an intelligence test at all", and such an explanation would be inappropriate. Rather, the ideal response shown informs the user why his preference is incompatible with the higher goal of determining a good diagnosis. Our model is capable of responding to such requests for the comparison of alternatives. The difficult problem does remain, however, of determining the user's intent in asking the question.

6.4. Conclusion

We have argued that in generating natural language explanations or responses an expert system must adhere to the pragmatic discourse conventions followed by human experts, particularly those that relate to cooperative behavior. In a task oriented context, user's of human experts may expect the expert to (i) recognize what goals and plans for attaining those goals they are attempting to achieve and respond to questions regarding those goals and plans appropriately, (ii) respond in terms of a better plan if the recognized one is either not the best plan or inappropriate for attaining the user's perceived goals. The use of a natural language explanation facility for a computer system that is *expert* will imply to the user that the responses adhere to the normal expectations of cooperative expert behavior. Not meeting these expectations may well confuse and mislead the user and will certainly not provide the best explanation or response.

Joshi (1984a, 1984b), presents examples where an advice-giving system should anticipate the possibility of the user drawing false conclusions from its response because of violated expectations and hence should alter or expand its response so as to prevent this from happening. Joshi's approach involves only the immediate goals of the user's query and does not present a general method of computing these responses.

Building on this research, we describe a model of the user based on a tracking and representing the user's goals, plans for achieving goals, and preferences among these goals and plans. An algorithm is presented which uses this model, together with the immediate goals of the query, to generate cooperative, helpful responses. By reasoning about the user's goals and plans, the algorithm can generate and present better alternatives to a user's task related query goal. Moreover, the algorithm can recognize several classes of direct responses which may violate the user's expectations of cooperative expert behavior and thus modifying the response to avoid misleading or confusing the user.

Our implementation has supported the claim that the approach is useful in an expert system environment where the user and the system work cooperatively towards common goals through the dialogue and the user's utterances may be viewed as actions in plans for achieving those goals. While our focus and examples have been on explanation in the context of expert advice giving systems, we feel this approach is applicable to a broad range of question types and to expert system explanation generation in general. We believe the present work is a small but never the less worthwhile step towards better and user-specific explanations from expert systems.

References

Allen, J. F., 1983, "Recognizing Intentions from Natural Language Utterances," in **Computational Models of Discourse**, Ed. M. Brady and R. C. Berwick, Cambridge: MIT Press.

Allen, J. F. and Perrault, C. R., 1980, "Analyzing Intention in Utterances," *Artificial Intelligence*, 15:143-178.

Austin, J. L., 1962, **How to Do Things with Words**, Cambridge: Harvard U.P.

Bromberger, S., 1965, "An Approach to Explanation," in **Analytical Philosophy**, Ed. R. J. Butler, Second Series, Oxford: Basil Blackwell.

Buchanan, B. G. and Shortliffe, E. H., eds., 1984, **Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project**, Reading, Mass.: Addison-Wesley.

Carberry, S., 1983a, "Tracking User Goals in an Information-Seeking Environment," *Proceedings of National Conference on Artificial Intelligence*, Washington, D.C.

Carberry, S., 1983b, "Understanding Pragmatically Ill-Formed Input," *Proceedings of AAAI-83*, Washington, D.C.

Carberry, S., 1985, "A Pragmatics-Based Approach to Understanding Intersentential Ellipsis," *Proceedings of ACL-85*, Chicago, Ill.

Cohen, P. R. and Levesque, H. J., 1985, "Speech Acts and Rationality," *Proceedings of ACL-85*, Chicago, Ill.

Cohen, R., 1983, "A Computational Model for the Analysis of Arguments," University of Toronto Technical Report CSRG-151, Toronto, Ont.

Cohen, R., 1984, "A Computational Theory of the Function of Clue Words in Argument Understanding," *Proceedings of COLING-84, 10th International Conference on Computational Linguistics*, Stanford, Calif.

Cohen, R., 1985, "The Need for Pragmatics in Natural Language Understanding," *Proceedings of CSCSI workshop in Theoretical Issues in Natural Language Understanding*, Halifax, Nova Scotia.

Cohen, R. and Jones, M., 1986, "Incorporating User Models into Expert Systems for Educational Diagnosis," submitted to *Computational Intelligence* special issue on AI approaches to education, August, 1986

Davis, R., and Lenat, D. B., 1982, in **Knowledge-Based Systems in Artificial Intelligence**, Toronto: McGraw-Hill.

Garfunkel, A., 1981, **Forms of Explanation**, Yale University Press.

Grice, H. P., 1975, "Logic and Conversation," in **Syntax and Semantics 3: Speech Acts**, Ed. P. Cole and J. Morgan, New York: Academic Press.

Hayes-Roth, F., Waterman, D. A., and Lenat, D. B., eds., 1983, **Building Expert Systems**, Reading, Mass.: Addison-Wesley.

Hempel, C. G., 1965, "Scientific Explanation," in **Aspects of Scientific Explanation**, Free Press.

Jones, M. and Poole, D., 1985, "An Expert System for Educational Diagnosis Based on Default Logic," *Proceedings of the Fifth International Conference on Expert Systems and Their Applications*, Avignon, France.

Joshi, A., Webber, B., and Weischedel, R., 1984a, "Living up to Expectations: Computing Expert Responses," *Proceedings of AAAI-84*, Austin, Tex.

Joshi, A., Webber, B., and Weischedel, R., 1984b, "Preventing False Inferences," *Proceedings of COLING-84, 10th International Conference on Computational Linguistics*, Stanford, Calif.

Kaplan, J., 1983, "Cooperative Responses from a Portable Natural Language Database Query System," in **Computational Models of Discourse**, Ed. M. Brady and R. C. Berwick, Cambridge: MIT Press.

Kindersley, S., 1986, "A Natural Language Interface to a Diagnosis System: A Study of WUP, the Portablilty of Shadow, and the Needs of the CGD Project," M. Math Thesis, University of Waterloo, Waterloo, Ont.

Kowalski, R., 1979, **Logic for Problem Solving**, New York: North Holland.

Litman, D. J. and Allen, J. F., 1984, "A Plan Recognition Model for Subdialogue in Conversations," University of Rochester Technical Report 141, Rochester, N.Y.

Mann, W. C. and Thompson, S. A., 1983, "Relational Propositions in Discourse," Information Sciences Institute Report RR-83-115, Marina del Rey, Calif.

Matthews, R. J., 1981, "Explaining and Explanation", *American Philosophical Quarterly*, 18:71-77.

McCoy, K. F., 1984, "Correcting Object-Related Misconceptions: How Should The System Respond?" *Proceedings of COLING-84, 10th International Conference on Computational Linguistics*, Stanford, Calif.

McKeown, K. R., 1982, "The Text System For Natural Language Generation: An Overview," *Proceedings of ACL-82*, Toronto, Ont.

McKeown, K. R., 1984, "Natural Language for Expert Systems: Comparisons with Database Systems," *Proceedings of COLING-84, 10th International Conference on Computational Linguistics*, Stanford, Calif.

McKeown, K. R. and Derr, M. A., 1984, "Using Focus to Generate Complex and Simple Sentences," *Proceedings of COLING-84, 10th International Conference on Computational Linguistics*, Stanford, Calif.

McKeown, K. R., Wish, M., and Matthews K., 1985, "Tailoring Explanations for the User," *Proceedings of IJCAI-85*, Los Angeles, Calif.

McLeod, J. and Jones, M., 1985, "The Development of an Expert System for Computer-Guided Diagnosis of Children's Learning Problems: Some Emerging Problems," *Proceedings of International Conference on the Computer as an Aid for Those with Special Needs*, Sheffield, England.

Mercer, R. E. and Rosenberg, R. S., 1984, "Generating Corrective Answers by Computing Presuppositions of Answers, not of Questions," *Proceedings of CSCSI-84*, London, Ont.

Nielsen, K., 1979, "Reason and Sentiment," in **Collection Philosophica**, Vol. 13, Ed. T.F. Geraets, Ottawa: University of Ottawa Press.

Nilsson, N. J., 1980, **Principles of Artificial Intelligence**, Palo Alto, Calif.: Tioga Publishing Company.

Paris, C. L., 1985, "Description Strategies for Naive and Expert Users," *Proceedings of ACL-85*, Chicago, Ill.

Perrault, C. R., Allen, J. F., and Cohen, P. R., 1978, "Speech Acts as a Basis for Understanding Dialogue Coherence," *Proceedings of Theoretical Issues in Natural Language Processing (TINLAP-78)*, Urbana-Champaign, Ill.

Pollack, M. E., 1984, "Good Answers to Bad Questions: Goal Inference in Expert Advice-Giving," *Proceedings of CSCSI-84*, London, Ont.

Reiter, R., 1980, "A Logic for Default Reasoning," *Artificial Intelligence*, 13:81-132.

Sacerdoti, E. D., 1977, **A Structure for Plans and Behavior**, New York: Elsevier AI Series.

Searle, J. R., 1969, **Speech Acts: An Essay in the Philosophy of Language**, Cambridge: Cambridge U.P.

Swartout, W. R., 1983, "XPLAIN: A System for Creating and Explaining Expert Consulting Systems," *Artificial Intelligence*, 21:285-325.