CUCS-85-83

User-Oriented Explanation for Expert Systems

Kathleen McKeown
Department of Computer Science
Columbia University
New York, NY 10027

Table of Contents

1 Project Summary

In this project we are concerned with extending the range and effectiveness of explanation facilities currently available in knowledge-based expert systems by incorporating the full power of natural language techniques. As expert systems become larger and more sophisticated, it is important that they be able to communicate their reasoning effectively to their users, whether they be naive users, domain experts, or system maintainers, thus allowing them to evaluate advice the system provides. One main problem with current explanation facilities is the inability to adequately tailor an explanation to a particular user of the system. In this project, we are examining the effects of the user and discourse situation on the type of explanation provided.

In particular, we are interested in developing capabilities for providing re-explanations when the user is dissatisfied with an explanation, for providing explanations in an appropriate level of detail for a given user, for adapting the explanation to the point of view the user has taken, and for using a persuasive strategy that the user is likely to accept. These efforts require the development of a semantically rich representation of the domain, capable of supporting abstractions and different points of view; the investigation of strategies that are effectively used for justification and causal explanation; and a development of the influences on level of detail, strategy selected, and point of view. We expect the results of this research to be both practical and theoretical: they will improve the range of man-machine communication currently available, making expert systems more accessible to their users, and will improve our understanding of how appropriate explanations can be produced.

2 Introduction

Expert systems have developed to the point where they are currently being successfully used in a variety of fields including medical diagnosis and computer configuration, among others. One of the key components of an expert system has always been an explanation facility. Explanation has been viewed as crucial in order that experts and users can understand how and why a system produced its result and can therefore evaluate whether the result is acceptable or not. More sophisticated existing explanation facilities are capable of describing the principles behind the reasoning done by the system.

While current explanation facilities have addressed some important issues in what kind of knowledge is needed to provide acceptable explanations, many issues in explanation for expert systems remain unaccounted for. As attempts are made to commercialize expert systems, deficiencies in current explanation facilities will be felt in two ways

- The transfer of existing explanation components to new systems and domains is made difficult by their design
- Inherent limitations make current explanations inadequate for some users in various situations

Our research addresses these issues by adapting *language generation* techniques to reduce the effort required to develop an explanation component for a new domain and by proposing several new areas of study to remove specific limitations inherent in existing systems. In particular, existing systems are limited by their inability to adjust their text to a specific user and the situation in which the explanation was requested, both of which can critically affect the kind of explanation that is required. This illustrates the need to identify the dimensions along which an explanation can and ought to vary as well as the influences on this variation. These research goals are described in more detail after presenting the current state of the art in explanation.

3 Existing Explanation Capabilities

The problem of generating explanations has been addressed both by researchers interested primarily in expert systems as well as researchers primarily interested in language generation. The most recent advances in both of these areas are briefly summarized here.

3.1 Expert System Approaches

Early work in explanation for expert systems is typified by the approach taken in MYCIN [SHORTLIFFE 76]. MYCIN uses backward-chaining techniques to arrive at a solution. Thus, at any time the system has available a hierarchical goal structure indicating the top-level goal and sub-goals it is currently working on. An explanation is provided by doing simple transformations on the goal structure (i.e., program trace). This is done by ascending the goal structure to describe *why* the system is pursuing its current goal and by descending the structure to describe *how* the system achieved its current goal.

The actual text of the explanation is provided by attaching *templates*, canned English phrases with slots, to each of the system's rules or goals. The system produces a *why* explanation by ascending the goal structure retrieving the template associated with each goal, filling its slots with the English translation of instantiated variables, and adding the English sentence (or phrase) to the explanation produced so far. The order in which sentences are added to the explanation is thus controlled by the ascent of the goal structure. Note that templates must be produced by hand ahead of time and care must be taken that they can be juxtaposed without problems to form a full English explanation. The main effort in designing such a system goes into the writing of the templates. This effort must be repeated when moving to a new domain. Moreover, consistency between what the system does and what it says is not guaranteed since rules representing goals can be changed independently of their templates.

While this technique works reasonably well to provide descriptions of what the program is doing, it is not adequate for providing justifications for why the program chose its course of action. That information is lost when the rules embodying the expert system were initially written (i.e., it is used to decide which rules are necessary but is not encoded in the rules themselves). Furthermore, explanation produced in this way is likely to include steps performed by the program which the physician/expert would normally perform unconsciously if at all (e.g. *I set the factor of reduction due to myxedema to 0.67*). Swartout [SWARTOUT 79] addressed this problem by developing a system that automatically produces the expert system and which records the domain principles used to produce the code. These principles can then be used to provide justifications for why the system took its course of action. Swartout also attached ***viewpoints*** to each rule and principle specifying whether it is appropriate for an expert or a programmer, thus allowing the system to delete detailed steps that are not necessary for the physician to see.

Swartout used ***direct translation*** of the underlying goal structure to produce the actual text of an explanation. Direct translation makes use of a simple grammar and dictionary and alleviates some of the problems of the template approach. Note, however, that the order and content of the explanation so produced is still controlled by the underlying goal structure, augmented now by the domain principles and viewpoints.

Additional work on the problem of knowledge needed for explanation was done by Clancey [CLANCEY 79]. He also found that an augmented knowledge base was needed in order to provide adequate explanations. Since he was interested in tutorial expert systems, he showed how tutorial strategies for medicine could be represented distinctly and used for explanation. The need for an adequately rich domain model incorporating levels of knowledge over and above that required to arrive at a solution continues to be a theme of research in expert system explanation (e.g. [COHEN 83], [PATIL 81]). We feel that these efforts provide a necessary starting point for further work in explanation and plan to use at least as detailed a representation as these works.

3.1.1 Limitations

Although some research in the expert system tradition has extensively examined the problem of what knowledge is needed for explanation, systems have not taken full advantage of the natural language techniques currently available for generation. Systems have either used direct translation of the underlying knowledge structure or templates to produce explanations. Knowledge about language and strategies for organizing explanations are not exploited in these schemes. This means that the quality of the explanation is determined by the structure of the underlying knowledge base. Note, furthermore, that since explanation so produced depends primarily on the underlying knowledge base and code, the same explanation will be produced for the same problem regardless of who is using the system and in what situation they request an explanation.

3.2 Language Generation Approaches

Research in language generation has dealt with two main issues: determining the content and the textual shape of what's to be said and translating a pre-determined message into English (see [MANN 83] for a summary of the state of the art in language generation). Systems which address the latter of these two issues are fairly well developed now. Most generation systems use a syntactic grammar of English and a reasonably developed dictionary to select appropriate syntactic structures and words to use rather than resorting to any form of canned text, including templates. This gives the system much more flexibility than any of the canned approaches allow since all phrases of explanations do not have to be constructed by hand beforehand, unforeseen interactions between juxtaposed phrases are handled by the grammar, and the exact form of the explanation can be determined dynamically upon its production. A few of the *direct translation* expert explanation systems (notably [SWARTOUT 81]) use a limited form of this technology. While a number of issues remain to be solved, particularly in the area of the effect of the user on linguistic choice, many systems (e.g., [MCDONALD 80], [MANN 83], [MCKEOWN 82A], [APPELT 81]) have demonstrated that this approach is feasible. We plan to take advantage of this technology and the flexibility that it allows in our proposed research.

More recently there have been a number of efforts (e.g., [WEINER 79], [COHEN 81], [MCKEOWN 82A], [REICHMAN 81]) to formalize possible strategies for text organization through analysis of naturally occurring texts or dialogues. This approach has proven successful, since it allows the generation process to structure information in the text differently than it is stored by using communicative strategies that people use effectively. Of these efforts, Weiner's and Cohen's have been aimed specifically at structures for presenting reasoning.

Planning mechanisms for language generation have been developed by [APPELT 81] and [COHEN 81]. This research investigates formalisms that are adequate for reasoning about the user's beliefs and goals in order to determine appropriate speech acts and utterances. Their efforts have concentrated on demonstrating how the planning mechanism can be used to take these criteria into account, developing the capability for reasoning about the user. This is an important facility for taking the user into account.

3.2.1 Limitations

More formal and detailed strategies need to be developed for presenting reasoning. Current work on textual strategies for presenting reasoning uses very simple combinations of possible techniques. Furthermore, there has been very little work done on what it means to be a *reason* or *evidence* for a *statement*, relations that are frequently used as kernels of explanation strategies. What counts as a good reason or as adequate evidence? Our work on points of view is intended to answer some of these questions. In addition, research on integrating strategies for organizing a text with formalisms that allow consideration of the user's beliefs and goals remains an open issue.

4 Our Previous Work

Our research to date has focused on the development of a language generation system that follows experimentally determined guidelines for effective communication. The major results include the development of a theoretical framework for dynamically determining what information is relevant for a given question and how to appropriately organize that information (see [MCKEOWN 82A], [MCKEOWN 82B], [MCKEOWN 83]). This framework is based on an analysis of the effects of discourse structure and focus constraints on the generation process. A computational treatment of rhetorical devices has been developed which is used to guide the generation process. Previous work on focus of attention (e.g., [GROSZ 77], [SIDNER 79]) has been extended for the task of generation to constrain what can be said next to that which ties in most appropriately with the previous discourse

This work on generation has been done as part of a natural language interface to a database system. The implemented system, TEXT, generates responses of paragraph length to questions about database structure. Three classes of questions have been considered: questions about information available in the database, requests for definitions, and questions about the differences between database entities.

The use of two interacting mechanisms encoding knowledge about discourse structure and focus of attention constitutes a departure from earlier generation systems. One of the main advantages is that instead of simply tracing the knowledge base to produce text, the system uses communicative strategies well founded on language principles to organize and effectively convey information represented in the knowledge base. This means that the same information may be described in different ways on different occasions and that some of the burden is taken off the design of the knowledge base since generation of appropriate text no longer depends on the knowledge base structure.

4.1 Application to Expert System Explanation

This approach could be applied to expert systems where the generation of explanation has typically been achieved by tracing the knowledge base or the current goal structure. The order of an explanation would be controlled by the communicative strategy used. The goal structure (with pointers into the augmented knowledge base) would still be used as the source of information for the explanation. The strategy, however, would dictate when to extract each piece of information from the goal structure to include in the explanation. Strategy may also affect the content of an explanation since a particular piece of information in the goal structure may not be relevant, given the justification strategy being used. This approach divorces the generation process from the design of the knowledge base. To generate a different explanation of the same result, a different strategy can be mapped onto the goal structure; the knowledge base need not be changed.

The communicative strategies, focusing principles, and grammar developed in the TEXT system would increase the generality of explanation for expert systems thereby making it easier to move the system to a new domain. Further work is needed, however, to improve the quality of explanations produced. In neither my own work nor in expert system explanation has anything more than a static, generic user been taken into account. This is currently needed in order to more fully meet the needs of expert system users.

5 Research Goals

Our aim is to develop the capacity for generating explanations that take the user into account. We define *user-oriented explanation* as having four main components:

1. The ability to provide re-explanations in response to a user's dissatisfaction with a given explanation. A user may not understand or agree with the first explanation s/he receives.

2. The ability to automatically choose the level of detail that is appropriate for the current user in different discourse situations. Each user, knowing different facts by virtue of previous discourse, will require a different amount of detail to understand an explanation.

3. The ability to use different justification strategies to explain the same reasoning differently for different users. A single justification strategy will be less convincing for some users than for others.

4. The ability to justify the same conclusion from different points of view. Dependent on the user's perspective (which may derive from his/her goals or focus), different information will be more or less relevant to the explanation.

All of these components have in common the assumption that there are many ways to explain a single phenomena or belief; they identify some of the dimensions along which explanation can vary. Furthermore, these are features which have been shown to be needed in expert systems [POLLACK 82] if they are to prove to be acceptable to their users.

While we aim to explore the ways in which explanation can vary in order to endow a system with the capacity for providing different explanations, we would also like the system to be able to choose the most appropriate explanation for a given discourse situation and user. This will require an examination of the influences on each of these dimensions with the goal of developing a framework for generating the explanation that is most suitable. Finally, this research will ultimately require additional work in the area of interpretation of natural language, since the generation tasks described above place demands on the analysis component to identify which situations and influences are present so that a tailored explanation can be generated. Each of the areas identified here are described in more detail below.

5.1 Re-explanation

Indication on the part of the user that an explanation was either not understood or did not meet his/her demands requires that a different explanation be generated. Clearly, generating the same explanation a second time will not bring the system any closer to satisfying the user. The ability to provide re-explanations is viewed as a precursor to implementing the initial selection of the correct dimension of explanation for a particular user. In requesting a re-explanation, the user often specifies the reason for dissatisfaction with the initial explanation. This will allow the system to select the appropriate dimension along which to change the explanation. By examining discourse situations in which an explanation was judged unacceptable by a user, it will be possible to develop a set of criteria for selecting the correct variation on first try. The ability to provide different explanations upon request will always be necessary, however, as a system which is unable to do so cannot be said to be sensitive to the needs of its users.

Re-explanations may be required to vary along any of the three dimensions identified above in addition to cases where the user rejects assumptions made by the system. Examples of these four classes of re-explanation are shown below to illustrate the system behavior towards which we are aiming. The examples shown below are constructed examples within a student-advisor domain. *U* identifies the user/questioner and *E* the expert/answerer

1. **User rejects or restates assumptions:**
   U: Will I be able to graduate by my senior year?
   E: If you take 3 courses each semester next year, then you can take 2 courses each semester your senior year and you'll be through.
   Restatement of constraints:
   U: But the thing is, I have to take data structures next semester and that means I can only take 2 courses since every other course requires it except for assembly language.
   New answer:
   E: Then you can take 3 courses one semester your senior year.

2. **Rejection of point of view:**
   U: Will I be able to complete the Computer Science major by my senior year?
   Gloss: Answer views process of major completion as filling a sequence of semester slots.
   E: If you take 2 CS courses per semester from now on you'll have no problem
   Gloss: User indicates s/he views process as a sequence of states by which one has completed certain steps.
   U: But everyone else I know decided on their major as a freshman.
   Gloss: Answer is re-oriented towards user's current state
   E: You've taken all the courses you should have taken by your sophomore year, so you'll be able to finish.

3. **Rejection of Reasoning Strategy:**
   U: Should I take data structures this summer?
   Support with evidence: Answer argues by virtue of the norm
   E: Yes, students usually take it in the summer if they didn't take it their sophomore year.
   Gloss: User rejects advice and opens up a second possibility
   U: But why can't I take it next year?
   Case Analysis: Answer considers consequences of 2 hypothetical cases
   E: If you take data structures this summer you'll be able to take some of your electives in the fall and fundamental algorithms in the spring. But if you wait until the fall to take it, you won't be able

to take any other CS courses in the fall and you'll fall behind in your major.

4. **Re-explanation which provides more detail:**
   U: Now I'll be able to finish my major next year?
   E: Absolutely, you only have 4 more courses to take.
   Gloss: User indicates he is unsure of the answer, needing more detail
   U: And I can fit them in?
   Gloss: Answer provides more detail on exactly how the user can complete the major
   E: You'll have to take 3 the second semester: fundamental algorithms, computability, and natural language processing are only offered in the spring. You can take artificial intelligence in the fall. But that's a reasonable plan for the year.

5.2 Level of Detail

A request for an explanation may be satisfied in several different ways simply by varying the amount of detail provided. Level of detail may be varied by level of abstraction (from more general to more specific) as well as by type of detail provided. Examples of variations along these lines are shown in (1)-(3) below.

1. **Most General:**
   U: Why should I take data structures?
   E: Because it's a requirement for most other Computer Science courses.
2. **More Specific:**
   E: Because it's a pre-requisite for almost all upper-level electives (such as Artificial Intelligence or Operating Systems) as well as many of the required courses.
3. **Different Type of Detail:**
   E: Because all other courses depend on you being knowledgeable about the types of data structures that are typically used in Computer Science such as trees, records, lists and queues.

5.3 Justification Strategy

There are many ways in which to arrive at the same conclusion (i.e., many different proof strategies) and these correspond as well to different methods for explaining the conclusion (see [WEBBER 82] for a preliminary analysis of these). In addition, there are different strategies for structuring proofs and reasoning in natural language (methods for doing so have been explored by [WEINER 79], [COHEN 81], and [REICHMAN 81]). The selection of an appropriate strategy in addition to the development of additional strategies is part of the work we plan to pursue

5.4 Point of View

Very often the content of an explanation or the point of view from which it is presented changes from one discourse situation to another, even when the advice presented is the same For example, if a user has specified interests or concerns of particular importance to him/her, the expert can provide an explanation that conforms to that point of view An example of four different justifications of the same answer from different points of veiw are given below We plan to investigate uniform strategies for identifying and accessing different points of view in a general enough manner that this can carry over to other domains

Question:

Should I take both discrete and data structures next semester?

Answers:

State Model: *What should be completed at each state in the process?*

Yes, you usually take them both first semester sophomore year.

Semester Scheduling: *How can courses be fit into schedule slots?*

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.

Requirements: *How do courses tie in with requirement sequencing?*

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

Workload: *How do courses achieve balanced workload?*

Yes, they complement each other and while data structures requires a lot of programming, discrete does not.

5.5 Influences on Explanation

Clearly, an analysis of the factors that trigger selection of an explanation dimension is critical for use of this capability and this is an endeavor that is currently underway. A preliminary analysis indicates that factors affecting level of detail would at least include the following (further work is needed to identify factors influencing other dimensions):

1. The user's level of expertise. A user comes to a system with apriori knowledge on the subject in question. The system's knowledge of that level (whether deduced from interaction or explicitly stated) will influence how much it should say. Note that this is not a simple influence. An expert may in certain situations be able to handle more detail than a novice.

2. The past discourse: What the user has learned through the past discourse influences level of detail since previous discussion of a subject may mean that less can be said about it in a current response. What the system has learned through the past discourse affects level of detail as well: the user's acceptance of detail or request for detail may indicate to a system that it can provide a particular type of detail without being asked.

3. The user's overall goal in interacting with the system. Whether the user is using the system, for instance, to quickly retrieve a specific fact or to learn about or from the system will require different levels of detail.

4. The user's specific goal in asking a particluar question: If the user's question is only one step towards acquiring the information necessary for a higher level goal, that goal may dicate how much information is required.

5. Feedback from the user: While the goal of this research is to anticipate the user's needs for detail before s/he states them explicitly, in actual converation, people often do explictly state that they have absorbed information and are ready for more (e.g., backchannel noises such as "um-hum") or that they have not understood. Such feedback can also be used in a system.

5.6 Interpretation

The need to identify which influences are present in order to generate the appropriate explanation places demands on the interpretation component. Since we would like the system to choose an explanation that is appropriate for a given discourse situation, some component of the system must be able to identify the discourse features that are present and bear on explanation. This brings the expert system user interface closer to a more flexible natural language interface than it currently is. While our emphasis is on the generation of explanations, the need to demonstrate the feasibility of generating user-oriented explanations may require some work in the area of natural language interpretation.

6 Relation to Other Work at Columbia

The proposed research described here is closely tied to research currently being done by Stolfo (e.g., [STOLFO 83]) on expert systems. In fact, Stolfo is also interested in explanation and has been exploring methods for making forward-chaining rule-based systems more amenable for explanation (explanation is easier in a backward-chaining systems). Our planned collaboration includes the following features: merging of expert system technology with language generation techniques, increased range of expertise with which to deal with unforeseen problems, interaction between graduate students, and sharing of software systems and tools.

We expect to mutually profit from our collaboration: Stolfo's extensive experience in expert system technology and my experience with language generation issues provide us with a solid base for further research on explanation for expert systems, an area which ultimately requires solutions from both fields.

7 Plans for Carrying Out Proposed Research

Our approach to developing user-oriented explanation will follow four stages of research comprising analysis of naturally occurring dialogues to observe both the extent of explanation variation and the effects on appropriate choice, formalization of the analysis results for computational use, a small prototype implementation to demonstrate the validity of the approach, and finally, the incorporation of the results in a larger expert system. Initially, we are primarily interested in the first three stages as this is where the bulk of the theoretical effort must go.

We have already begun to develop the capacity for generating different explanations of the same phenomena by investigating methods for representing different points of view in the knowledge base. Our next step will be to formalize reasoning strategies, incorporating the work that has already been done in this area. This will form the basis for variation in explanation. Dialogues will then be analyzed to determine the discourse situations in which a user is and is not satisfied with a given explanation. Thus, we first plan to develop the ability to provide different explanations and following that, the ability to tune the explanation for the current user.

Implementation of the prototype system will be done in a domain in which the planning that must be done is not extremely complex so that we may focus on the explanation aspects of the system. We have chosen a student-advisor domain because it is a domain in which the planning can be kept relatively simple and yet there is a real need for communicating at different levels of detail and for providing re-explanations to students who may be dissatisfied with an explanation (for example, why they cannot take a course), may simply want to talk at length about a course of action, and may want to explore alternate solutions to a problem.

While this domain lends itself to the problems we have chosen to study, we do not view our results as being applicable to this domain only. In order to demonstrate domain independence and validity in a larger, real-world expert system environment, we plan to implement results in a larger system. Possibilities for this system include some of the computational environments that are currently being used at Columbia such as ACE or MVS.

8 Conclusions

The development of explanation facilities continues to be an important problem in the expert system area. In this proposal, we have outlined some ways in which existing explanation systems are deficient in regard to their users and have proposed methods to deal with this problem. User-oriented explanation is a crucial next step if expert systems are to be accepted by the community at large.

9 References

[APPELT 81] Appelt, D.E., Planning natural language utterances to satisfy multiple goals, Ph.D. dissertation, Stanford University, Stanford, Ca., 1981.

[COHEN 78] Cohen, P., On knowing what to say: Planning speech acts, Technical Report No. 118, University of Toronto, toronto, 1978.

[COHEN 81] Cohen, R., Investigation of processing strategies for the structural analysis of arguments, in *Proceedings of the 19th Annual Meeting of the Association for Computational Linguistics*, Stanford, Ca., June 1981, pp. 71-6.

[COHEN 83] Cohen, P.R. and M.R. Grinberg, A theory of heuristic reasoning about uncertainty, *The AI Magazine*, Vol. 4, No. 2, Summer 1983, pp. 17-24.

[CLANCEY 79] Clancey, W.J., Tutoring rules for guiding a case method dialogue, *International Journal of Man-Machine Studies 11*, 1979, pp. 25-49.

[GROSZ 77] Grosz, B. J., The representation and use of focus in dialogue understanding. Technical note 151, SRI International, Menlo Park, Ca. (1977).

[MANN 83] Mann, W., An overview of the NIGEL text generation grammar, *Proceedings of the 21st Annual Meeting of the Assocation for Computational Linguistics*, Cambridge, Mass., 1983, pp. 79-84.

[MCDONALD 80] McDonald, D.D., Natural language production as a process of decisions making under constraint. PhD. Dissertation, MIT, Cambridge, Mass. 1980.

[MCKEOWN 80] McKeown, K.R., Generating relevant explanations: natural language responses to questions about database structure, *Proceedings of the First Annual National Conference on Artificial Intelligence*, Stanford, Ca., 1980, pp. 306-9.

[MCKEOWN 82A] Mckeown, K.R., Generating natural language text in response to questions about database structure. Ph.D. Thesis, MS-CIS-82-05, University of Pennsylvania, Philadelphia, 1982.

[MCKEOWN 82B] McKeown, K.R., The TEXT system for natural language generation: an overview, *Proceedings of the 20th Annual Meeting of the Association for Computational Linguistics*, Toronto, Ontario, Canada, 1982, pp. 113-20.

[PATIL 81] Patil, R.S., Szolovits, P., and W.B. Schwartz, Causal understanding of patient illness in medical diagnosis. *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, Vancouver, British Columbia, August 1981, pp. 893-9.

[POLLACK 82] Pollack, M., Hirschberg, J., and B. Webber, User participation in the reasoning processes of expert systems, *Proceedings of the National Conference on Artificial Intelligence*, Pittsburgh, Pa., August 1982, pp. 358-61.

[REICHMAN 81] Reichman, R., Plain speaking: a theory and grammar of spontaneous discourse. Ph.D. Thesis, Harvard University, Cambridge, Ma., 1981.

[SHORTLIFFE 76] Shortliffe, E.H., *Computer-Based Medical Consultations: MYCIN*, New York: Elsevier/North-Holland, 1976.

[SIDNER 79] Sidner, C.L., Towards a computational theory of definite anaphora comprehension in English discourse, Ph.D. dissertation, MIT, Cambridge, Mass., 1979.

[STOLFO 83] Stolfo, S.J., Knowledge engineering: Theory and practice, Technical Report, Computer Science Dept., Columbia Univ., New York, N.Y 1983.

[SWARTOUT 81] Swartout, W.R., Producing explanations and justifications of expert consulting programs, Technical Report MIT/LCS/TR-251, MIT, Cambridge, Mass., January 1981.

[WEBBER 82] Webber, B. and A.K. Joshi, Taking the initiative in natural language data base interactions: Justifying why, Technical Report, University of Pennsylvania, Philadelphia, Pa., 1982.

[WEINER 79] Weiner, J.J., BLAH, A system which explains its reasoning, *Artificial Intelligence 15*. 1980, pp. 19-48.