Preprint of: Robert Kass and Tim Finin, The need for user models in generating expert system explanation. International Journal of Expert Systems, v1, n4, pp 345-375, October 1988.

Department of Computer & Information Science

Technical Reports (CIS)

University of Pennsylvania *Year* 1988

The Need for User Models in Generating Expert System Explanations

Robert Kass
University of Pennsylvania

Timothy Finin
Unisys, Inc.

This paper is posted at ScholarlyCommons.
http://repository.upenn.edu/cis_reports/585

Preprint of: Robert Kass and Tim Finin, The need for user models in generating expert system explanation. International Journal of Expert Systems, v1, n4, pp 345-375, October 1988.

THE NEED FOR USER MODELS IN GENERATING EXPERT SYSTEM EXPLANATIONS

Robert Kass
Tim Finin

MS-CIS-88-37
LINC LAB 114

Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania
Philadelphia, PA 19104

June 1988

To appear in a forth coming issue of the "International Journal of Expert Systems"

Acknowledgements: This research was supported in part by Digital Equipment Corporation, DARPA grant NOOO14-85-K-0018, NSF grants MCS-8219196-CER, IRI84-10413-AO2 and U.S. Army grants DAA29-84-K-0061, DAA29-84-9-0027.

Preprint of: Robert Kass and Tim Finin, The need for user models in generating expert system explanation. International Journal of Expert Systems, v1, n4, pp 345-375, October 1988.

The Need for User Models in Generating Expert System Explanations

Robert Kass*
Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104-6389
(kass@linc.cis.upenn.edu)

Tim Finin
Paoli Research Center
Unisys
Paoli, PA 19301
(finin@prc.unisys.com)

June 2, 1988

Abstract

An explanation facility is an important component of an expert system, but current systems for the most part have neglected the importance of tailoring a system's explanations to the user. This paper explores the role of user modelling in generating expert system explanations, making the claim that individualized user models are essential to produce good explanations when the system users vary in their knowledge of the domain, or in their goals, plans, and preferences. To make this argument, a characterization of explanation, and *good* explanation is made, leading to a presentation of how knowledge about the user affects the various aspects of a good explanation. Individualized user models are not only important, it is pratical to obtain them. A method for acquiring a model of the user's beliefs implicitly by "eavesdropping" on the interaction between user and system is presented, along with examples of how this information can be used to tailor an explanation.

*This work was supported by grants from the Army Research Office and the Digital Equipment Corporation

Preprint of: Robert Kass and Tim Finin, The need for user models in generating expert system explanation. International Journal of Expert Systems, v1, n4, pp 345-375, October 1988.

Contents

Preprint of: Robert Kass and Tim Finin, The need for user models in generating expert system explanation. International Journal of Expert Systems, v1, n4, pp 345-375, October 1988.

1 Introduction

A distinctive feature of expert systems is the explicit representation of the reasoning and domain knowledge they use, enabling them to provide an *explanation* for the conclusions they reach. Unlike other decision systems, where the answer or recommendation is often presented without support, expert systems allow the user to explore the reasoning process that lead to the conclusion. In fact, explanation capabilities are frequently the most significant benefit provided by an expert system.

This paper examines the role of explanation in expert systems, and why user models are important to the generation of good explanations. The thesis is essentially this: when producing an explanation, a system makes assumptions about the knowledge of the user; if the system is designed to interact with a range of users whose domain knowledge varies, then explicit user models will be necessary, in order to generate good explanations. Furthermore, it is feasible to acquire such models as a user communicates with the system, and the models acquired in this way will be sufficient to tailor expert system explanations to individual users so that they will find the explanations more understandable.

The path taken to substantiate this thesis is somewhat long, occupying the body of the paper. First, the importance of an explanation facility for expert systems is discussed in section 2, arguing that the essential role of explanation in many expert systems justifies efforts to improve their explaining capabilities. Unfortunately, our understanding of explanation and the process of explaining within the Artificial Intelligence community is primarily intuitive. To provide a more solid basis for talking about explanation, material from the Philosophy of Science discussing explanations and their quality, augmented with a discussion of computational issues, is presented in sections 3 and 4. At this point, the need for knowledge about the user in producing good explanations will be evident. Section 5 considers the points in explanation generation where a user model is important, and how knowledge of the user's beliefs about the domain and reasoning knowledge of the system can influence explanations produced.

The role of user models in explanation generation is merely an academic exercise if such models cannot be acquired practically. Section 6 summarizes our work on acquiring knowledge of the user's beliefs about the system's domain and reasoning knowledge—knowledge used in the explanation generation process. These user model acquisition techniques build a user model implicitly, by "eavesdropping" on the interaction between system and user. Furthermore, the techniques are domain independent, enabling a general user modelling facility that can be used effectively in a range of systems with minimal customization. Section 7 illustrates how the acquisition rules acquire a model of the user's beliefs from a dialogue between the system and a user, and how these beliefs can be used to tailor an explanation so it is more understandable, with an extended example of a hypothetical investment advisory system.

2 The Importance of Explanation

An important feature of expert systems is their ability to explain their own reasoning. In summarizing a survey of physician's expectations and demands for computer-based consultation systems (reported in [36]) Buchanan and Shortliffe note:

> ...a program's ability to give explanations for its reasoning was judged to be the single most important requirement for an advice-giving system in medicine." [5, p. 603]

Weiner states that knowledge-based systems "...include some mechanism for giving explanations, since their credibility depends on the user's ability to follow their reasoning, thereby verifying that an answer is correct." [41, p. 19]. Explanation is thus a crucial feature of expert systems.

Preprint of: Robert Kass and Tim Finin, The need for user models in generating expert system explanation. International Journal of Expert Systems, v1, n4, pp 345-375, October 1988.

2.1 Why Explanation is Important

An explanation capability is important in an expert system for several reasons. The most common reason is the one noted by Weiner above: explanations are needed to *justify* a recommendation. An explanation can increase user acceptance of a recommendation by providing assurance that the recommendation is logical, or it can persuade a user that an unexpected recommendation is appropriate [39].

Explanations also help the user to recognize the limitations of an expert system. Neches, Swartout, and Moore [24, 25] have noted that system recommendations can confuse users who are unsure of the scope of system capabilities. An explanation facility enables the user to explore the system's reasoning to determine whether the system considered all the relevant facts, and reasoned with them appropriately.

Explanation can be an important aid to expert system development and maintenance, by providing a history of the reasoning steps taken by the system. MYCIN's explanation facility originated in commands to aid in debugging rules [5, p. 331–332]. Neches, Swartout, and Moore [24, 25] have exploited the relationship between explainability and maintainability in their Explainable Expert Systems (EES) approach, using declarative knowledge representations and an automatic program writer to produce systems that have good explanation facilities and are more maintainable than traditional expert systems.

Finally, an explanation facility can enable an expert system to instruct users about the system's domain. The GUIDON project [11, 10] exploited MYCIN's explanation capabilities to build an intelligent tutoring system for medical diagnosis. Even in conventional expert systems, an explanation capability may allow the system to provide a user with information he did not know, such as defining concepts relevant to the domain, as the CLEAR system does [31].

2.2 How Explanation Can Be Improved

Research on improving explanation has focused on two approaches: extending the range of possible explanations that an expert system can provide, and improving the quality of explanations produced. Conventional expert system explanations are limited to providing a description of the steps the system took in reaching a conclusions. For example, when MYCIN requests a piece of information, the user is allowed to type "WHY?" (interpreted to mean "How is this information useful?"), causing the system to produce an explanation based on the rule this goal appears in, and the goal the rule concludes about [9]. Such systems are capable of explaining what they did, but cannot justify those actions [34].

Clancey [9] and Swartout [34] have considered ways to extend the range of explanations an expert system can produce. Each has discovered that to produce explanations that address the intentions behind the system's actions, the strategic knowledge used by the system to reason about the domain must be separated from definitional and causal knowledge about the domain.

The second research issue, improving the quality of explanations produced, is the focus of this paper. Early work in this area includes the translation of formal proofs into English [8] and Weiner's BLAH system [41], which focused on how to organize and focus the information in an explanation. Wallis and Shortliffe [39], Sleeman [33], McKeown [21], and Paris [26] have presented methods for tailoring explanations according to user knowledge. The common emphasis of this approach is a focus on deciding what information an explanation should include, and how that information should be presented to benefit the user most.

Preprint of: Robert Kass and Tim Finin, The need for user models in generating expert system explanation. International Journal of Expert Systems, v1, n4, pp 345-375, October 1988.

3 What is an Explanation?

Since this paper focuses on explanation, and particularly on how to produce quality explanations, it is important to have a firm basis for discussion. Unfortunately, in expert systems research explanations are usually considered as simply the response an expert system makes to a "why?" question. Such a view is inadequate. Explaining is a type of human communication, one that expert systems are intended to mimic in their responses. Thus, it is important to characterize explanation and the explaining act. This section briefly discusses types of explanations, then presents a formal description of explanation and explaining taken from the Philosophy of Science. This analysis of explanation will serve as a basis for the following section, where the characteristics of good explanations are considered.

3.1 Types of Explanations

Explanations can take many forms, depending on the type of information they communicate. Perhaps the most familiar type of explanation is *scientific explanation*, where an argument is given to support a specific conclusion. Such arguments make take the form of deductive proofs from certain or uncertain premises (labeled *deductive-nomological* and *deductive-statistical* explanations respectively by Hempel [15]). Other types of scientific explanations may argue inductively (*inductive-statistical* explanations), or argue from the relation of a particular theory to the observed world (*statistical-relevance* explanations) [32].

Although the scientific style of explanation is the most obvious type of explanation, there are many others as well. Descriptions of objects or processes are often used to explain, such as explaining how to send an electronic mail message by describing the sequence of steps required to accomplish the action, or describing a telephone to explain how it works. In other cases, the explanation may take the form of an argument, but without the strict reasoning methods of a scientific argument, such as arguments that rely on analogy, examples, or an appeal to an authority.

Although current expert systems tend to give scientific-style explanations, any form of explanation may be appropriate. Since an explanation is used to communicate information to the user, systems should use the type of explanation that is most likely to be successful. In some cases, this may mean that the use of analogy or example is more appropriate than a complex scientific explanation. For example, the analogy between water flow and electricity is often used when explaining electrical properties. In other cases, the system may need to describe objects or processes in the domain, rather than argue about them. Thus, expert system explanations can, and should, take on a variety of forms depending on the explaining situation. A good explanation facility must be able to decide which form of explanation is most appropriate for a given situation.

3.2 Explaining

Having a typology of explanations is useful for understanding the range of circumstances where explanations might appear, and the range of behaviors that might be labeled "explaining." A typology is not sufficient, however, to enable one to point and say "That is an explanation!," or "That can't be an explanation!" Such a capability requires a specification of what conditions are necessary and sufficient for identifying an explanation. This section reformulates the issue of identifying explanations by first claiming that explanations are products of explaining actions. Furthermore, explaining is an illocutionary action [3], hence necessary and sufficient conditions for explaining must account for its illocutionary nature. These conditions are presented, along with further characteristics of explanation and explaining.

Preprint of: Robert Kass and Tim Finin, The need for user models in generating expert system explanation. International Journal of Expert Systems, v1, n4, pp 345-375, October 1988.

Explanation is the Product of Explaining

The concept of explanation is closely tied to the phenomenon of *understanding*, in that explanations aid the hearer's understanding of that which is being explained. However, everything that aids understanding is not an explanation. Suppose a person observes a billiard game, and by this observation comes to understand Newton's laws of motion—the billiard game did not explain Newton's laws of motion, rather, it happened to aid in a particular understanding event. On the other hand, having the form of an explanation (such as a logical argument form) does not make a statement an explanation—it may be wrong or universally misunderstood. Thus, although explanation involves understanding, there is more to it—the explanation must be presented *as* an explanation. In other words, an explanation is the result of an explaining action.

Explaining is an *illocutionary* action [3], an action performed with a particular intention in mind. Other types of illocutionary acts include warning, commanding, or promising.[1] To demonstrate this, consider a situation where a person, while reading the warning sign "Flammable Liquid: No Smoking" comes to realize that an exposed flame can cause a flammable liquid to explode. Certainly, the warning sign cannot be considered to explain the fact that exposed flames can cause flammable liquids to explode, because the sign was never *intended* to be an explanation. Thus, not only should a statement contribute to understanding to be considered an explanation, it should also be produced with the intention that it explain. Hence, explaining actions are illocutionary actions.[2]

Conditions for Explaining

Achinstein, in his book "The Nature of Explanation" [1] has presented a formal account of explaining, explanation, and what it means to be a good explanation, stating three necessary and sufficient conditions for a speaker S to explain "why p" to a hearer H by uttering a statement u expressing the proposition e.[3]

1. *Intentionality*: S utters u with the intention that H understand "why p."

2. *Correctness*: S believes that e is a correct answer to the question "Why p?"

3. *Instrumentality*: S utters u with the intention that u will produce the knowledge of "why p" in H.

The intentionality condition captures the illocutionary nature of explaining, while the correctness condition specifies that the speaker believes what he say *is* a correct answer. The instrumentality requirement is necessary to ensure that telling the hearer how to acquire the answer to "Why p?" (such as telling him where to look for the answer) does not count as the explanation of "why p."

It is enlightening to consider two features that are not necessary conditions for an explaining act. First, it is not necessary that the hearer recognize the explaining act. For example, H may not be listening when S utters u. The act is still the same, whether H realizes it or not, so it seems appropriate to omit H's realization of the act as a requirement for explaining. Similarly, explaining is not a perlocutionary act. If explaining were a perlocutionary act, an additional condition,

[1] Illocutionary acts are distinguished from locutionary acts, such as simple utterances, and from perlocutionary acts, which include the effects of the action on other agents. Thus commanding someone to stop is an illocutionary act. The actual statement "Stop!" is a locutionary act, while causing the person to stop is a perlocutionary act.

[2] Another reasonable possibility is to consider explanations to be perlocutionary acts. This issue will be discussed in the next section.

[3] This definition is taken from [1, pp. 16–18], but notation has been significantly changed, and in some cases technical aspects of the definition have been omitted.

Preprint of: Robert Kass and Tim Finin, The need for user models in generating expert system explanation. International Journal of Expert Systems, v1, n4, pp 345-375, October 1988.

concerning the effect of the explanation on the hearer, would be needed, such as the requirement that the hearer understand the explanation. Such a requirement would eliminate the possibility for failures when the speaker produces an explanation, but the hearer fails to understand.

An *explanation* is simply the product of an explaining act. More precisely, an explanation can be represented as an ordered pair consisting of a proposition and an act-type, so an explanation answering the question "Why p?" will be the pair "(e, explaining p)."

3.3 Understanding

An important point left open in the conditions for explaining is the meaning of the term "understand." Achinstein devotes a significant portion of his book to an often technical discussion of understanding, arriving at three conditions that define understanding.

The first condition is simple: for an agent to understand that e explains "why p," he must believe that e is a correct answer to the question.

Second, one does not understand "why p" in isolation, but rather in a certain *way*. In fact, an agent may know an explanation for why p is the case, but ask "Why p?" to learn an explanation of another sort. Thus, understanding an explanation includes the recognition that e provides a particular kind of answer to "Why p?."[4]

The last condition for understanding is the most difficult. It is not sufficient to know a proposition that serves as an answer to "Why p?," understanding also involves some notion of the *relation* between the answer and the question. For example, the classification of explanations in section 3.1 presents many types of relations between e and p. Just as one can explain q by analogy with p, an agent can understand q by analogy with p. Thus, the final condition for understanding is that an appropriate relation exists between e and "why p." This relation is summarized in Achinstein's notion of a *content-giving proposition*, a proposition that can be used with respect to "why p."

3.4 Contrast Classes

> *When Willy Sutton was in prison, a priest who was trying to reform him asked him why he robbed banks. "Well," Sutton replied, "that's where the money is." [13, p. 21]*

A further requirement for explanation has been noted by Garfinkel [13] and van Fraassen [37]. They observe that p, the event or situation being explained, is always distinguished from some set of alternatives, which they call p's *contrast class*. Van Fraassen claims this contrast class is an implicit part of any why-question. For example, in the anecdote above, the question "Why do you rob banks" could have several contrast classes, such as

Why do you rob *banks*? (That's where the money is.)
Why do you *rob* banks? (I don't like to work.)
Why do *you* rob banks? (Because my wife won't.)

and so on. The contrast class may be explicitly stated in a why-question, but it is usually implied by the context of the question and the current focus of the conversation. In either case, the question "Why p?" has an associated contrast class integral to the question. Thus, the task of explaining consists not simply of selecting an answer to present, but in selecting an answer from the correct contrast class, while denying that other members of the contrast class are a correct answer to the question.

[4] Achinstein formalizes this point by introducing the notion of *instructions* as constraints on an explanation—any explanation for "why p" must also satisfy some set of instructions I.

Preprint of: Robert Kass and Tim Finin, The need for user models in generating expert system explanation. International Journal of Expert Systems, v1, n4, pp 345-375, October 1988.

This notion of a contrast class is lacking in most current expert systems. Systems that explain by reciting the history of what rules where used do not keep track of alternatives, thus, they cannot argue why one sequence of rules was followed instead of another. However, Swartout's XPLAIN [34] and EES systems [35] do address this issue to some degree, in that the alternative methods for reaching a solution are recorded, allowing explanations of why one method was used rather than another.

3.5 Explaining and Justifying

Achinstein, Garfinkel, and van Fraassen are concerned only with characterizing scientific explanation, while the range of expert system explanations is not so limited. Thus, it is appropriate to distinguish two related activities that up to now have been lumped together under the term "explaining." *Explaining* will be used in the sense defined by Achinstein: having the intention to produce *knowledge* in the hearer. On the other hand, *justifying* is weaker than explaining, only intending to affect the *beliefs* of the hearer. Explanations and justifications are the result of explaining and justifying acts, respectively. Explanation deal with things that are true, while justifications are concerned with things that may not be true.

Fortunately, Achinstein's conditions for explanation can be applied to justifications as well, by relaxing the understandablity and correctness requirements. When explaining, S believes that he is giving a *correct* answer to "Why p?," while in justifying, S only believes his answer *supports* or *is evidence for* "why p." Likewise, in explaining, S intends H to *know* "why p," while in justifying S only intends that H accept e as support for why p. Formally, the necessary and sufficient conditions for performing a justifying act are the following.

1. *Intentionality*: S utters u with the intention that H accept e as support for "why p."

2. *Correctness*: S believes that e supports "why p."

3. *Instrumentality*: S utters u with the intention that e be accepted as support for "why p" by H.

In summary, explanation must be understood as the product of the illocutionary act of explaining. An explanation answers a why question, whether the question is explicit or implicit. In explaining, the speaker believes his statement answers the question "why p" (the question being explained), and intends that the hearer understand "why p" through his statement. Moreover, the question "why p?" must be considered in the context in which it is asked. An explanation answers a why question with respect to the space of possibilities associated (usually implicitly) with the question. In practical situations, a user model will be useful in producing an explanation by helping a system to determine *what* question the user is asking (or might ask), and what range of possibilities he considers to be potential answers.

4 What Makes an Explanation Good?

Characterizing explanations and explaining is not sufficient to enable one to begin building explanation facilities for expert systems. Although precise conditions for explaining have been presented, the space of possible explanations satisfying those conditions for any given question may still be quite large. An explanation generator must consider how to chose among the valid explanations possible, hence it requires some way of determining the *quality* of those explanations.

Preprint of: Robert Kass and Tim Finin, The need for user models in generating expert system explanation. International Journal of Expert Systems, v1, n4, pp 345-375, October 1988.

Just as section 3 examined the characteristics of (and conditions for being) an explanation, here, the features of *good* explanations are analyzed. Unfortunately, Achinstein resorts to vague terms such as "interesting" and "valuable" to characterize good explanations, terms that are as hard to characterize as "good." Thus, our discussion of explanation quality must go beyond Achinstein's treatment to identify specific characteristics of explanation quality that can be used to guide explanation generation.

A starting point for this analysis is Austin's communicative acts. In section 3, explaining was described as an illocutionary action—solely in terms of the speaker's intent. This characterization was necessary to account for the fact that explanations may vary in quality. When considering an explanation's quality, however, the expected perlocutionary effect of the explanation must be considered as well. This does not mean that the quality of an explanation depends on its actual success—an explanation can still be considered good, even if the hearer does not attend to it. Rather, the quality of an explanation should depend on how successful the explanation is expected to be, given its context.

This section presents three criteria for evaluating the quality of an explanation. Two (relevancy, and convincingness) affect the *content* of the explanation, while the third (understanding) primarily affects *how* that content is communicated.

4.1 Relevant Explanations

The first requirement of a good explanation is that it be relevant to the hearer's needs. To some degree this is covered by the requirements of explanation itself: to explain "why p?" the speaker must respond with respect to the implicit contrast class of the question. Thus, a question that expects an intentional answer cannot be explained by a causal answer. However, the relevance of a response goes beyond simply answering the question. Often agents have higher goals they wish to accomplish, obtaining a particular explanation may be a small step in achieving those goals. An explanation will be more relevant if it addresses the hearer's higher goals in addition to answering the immediate question.

Satisfying the user's goals is at the root of Achinstein's inclusion of "interest" and "value" as requirements for good explanation. An explanation is interesting if it addresses the user's goals in seeking the explanation, and valuable if it contributes to the accomplishment of those goals.

Producing relevant explanations is one aspect of cooperative behavior, described by Grice in his maxims of cooperativity [14]. Grice's maxims have been used extensively to guide research in the generation of cooperative responses, particularly in question answering systems. Two of these maxims have bearing on the relevance of explanations. The first, the maxim of relation, simply states "be relevant." This maxim summarizes the point made in this section, a relevant explanation is one that addresses the hearer's current goals. The second maxim, the maxim of quantity, goes a step further. The maxim of quantity says "Make your contribution as informative as necessary, but not more so." This maxim expresses the requirement that not only should a good explanation be relevant to the hearer's goals, it should address those goals as fully as possible. Furthermore, a good explanation will not include extraneous, irrelevant information that might be confusing to the hearer.

The relevancy criteria for good explanation affects the *content* of the explanation: what information is actually intended to be understood by the hearer after hearing the explanation. For an expert system, much of this information selection depends on the perceived goals of the user. This provides a partial method of evaluating the quality of explanations, in that one explanation is better than another if it enables the user to accomplish his task more quickly or with less effort. In producing good explanations, therefore, an expert system must attempt to maximize the likelihood

Preprint of: Robert Kass and Tim Finin, The need for user models in generating expert system explanation. International Journal of Expert Systems, v1, n4, pp 345-375, October 1988.

that the explanation it gives will satisfy the goals of the user.

4.2 Convincing Justifications

A justification presents an argument for belief, but the hearer may refuse to accept that argument. Thus, not only must good justifications be relevant and understandable, they should *convince* the user to believe what is being justified.

The convincingness of a justification depends on two things: the soundness of the justification itself, and the extent to which the hearer finds the justification acceptable. The soundness criterion affects the argument itself: a deductive argument is stronger than an inductive one, highly certain inferences are better than questionable ones, and scientific justifications are better than analogies or examples. In addition, for an argument to be convincing, the hearer must accept its premises and reasoning steps. Thus, it is better to argue based on facts the hearer believes, rather than facts the hearer is uncertain about, or does not believe at all. Likewise, a particular hearer may consider analogies to be perfectly acceptable arguments, meaning that an analogy from strong premises could be a more convincing justification for that hearer than a deductive argument from weak premises.

The convincingness criteria also affects the content of an explanation, since it helps determine what form of explanation to give, and what facts or arguments to include in that explanation.

4.3 Understandable Explanations

Not only must an explanation be relevant to the user, he must find it understandable. As with relevance, in order to produce good explanations, an expert system must strive to maximize the likelihood that the user indeed understands the explanation. Five features affect the understandability of an explanation: the first four (appropriateness, economy, organization, and familiarity) are concerned wih features of the explanation itself, while the fifth (processing requirements) considers understandability in terms of the effort on the part of the hearer. In fact, the processing required of a hearer to understand an explanation seems to be a primary criteria for judging the understandability of an explanation.

Appropriateness To be understood, a speaker must select a type of explanation the hearer is likely to understand. For example, in explaining how a light switch controls a light, one might give a physical-causal explanation, an analogy to water flow, or simply describe a sequence of events; depending on the knowledge the listener had of electricity. Paris, in studying the types of descriptions given by encyclopedias for children versus those for adults [26], discovered that significantly different forms of explanation are used for persons with different degrees of knowledge: for novices the explanations tend to be process-oriented, while for knowledgeable persons the explanations will tend to describe an object in terms of its properties, its parts, and things it is a part of. Thus, the kind of explanation appropriate for a user is dependent on his level of knowledge.

Economy Webber and Joshi [40] state that justifications from a knowledge base should be *succinct.* A succinct justification (or explanation) is one that does not provide more information than is necessary, and that provides this information with a minimum of words. Thus, with other things being equal, a short explanation is better than a long one.

Organization The *organization* of an explanation also affects its understandability: simple explanations are more likely to be understood than complex ones, and explanations that highlight

Preprint of: Robert Kass and Tim Finin, The need for user models in generating expert system explanation. International Journal of Expert Systems, v1, n4, pp 345-375, October 1988.

points of interest to the user will be more successful. Weiner [41] identified several organizational features affecting the complexity of an explanation, including: the amount the of detail used, the grouping of information, and the presence of structural information in the explanation. These issues have also been addressed by McKeown in organizing output in the TEXT system [20]. A well-organized explanation avoids large amounts of detail, collects related information together, and gives clues to help the user understand its structure.

Familiarity Another issue is the *familiarity* to the hearer of the content of the explanation. An explanation is more likely to be understood if it is expressed using terms the hearer is familiar with. Familiarity also affects the succinctness and complexity of an explanation, because unfamiliar terms or concepts will need to be explained for the hearer to understand the explanation. Thus, the use of unfamiliar terms in an explanation causes an explanation to be longer and more complex. On the other hand, if the hearer is familiar with portions of an explanation, they can be omitted. This happens frequently in logical deduction explanations, where reasoning steps are left out because the hearer is presumed to know them, and how they apply to the explanation.

Processing Requirements An issue that encompasses the features described above is the *processing required* on the part of the hearer to understand an explanation. These processing requirements can be divided into two categories: the amount of work the hearer must do to comprehend the explanation itself, and the amount of work required to infer the information the speaker intended to communicate by the explanation.[5] The issues of economy, organization, and familiarity affect the processing required of the hearer by determining the relative amount of work the hearer must do to comprehend the explaining statement, versus the work to infer the speaker's intended information. For example, an explanation may be well-structured and succinct, requiring very little work for the hearer to comprehend it, but the explanation may be indirect, causing the hearer to make further inferences to understand what the speaker was trying to communicate.

Processing requirements are important feature of explanation because they provide a single measure for understandability of explanations: explanations that require more processing to understand are more prone to be misunderstood. Thus, an expert system should seek to minimize the anticipated processing requirements of the user. Such a measure is attractive because it captures the intuition that a complete explanation, even though it is cooperative and correct, is not necessarily better than a short explanation that still addresses the user's goals.

To produce a good explanation or justification, issues of cooperativity, argumentation, and understandability must be addressed. The response must address the user's goals and preferences, be argued convincingly, and expressed in a manner that is likely to communicate the intended information. As discussed in the next section, explanation capabilities in each of these areas can be enhanced by the use of knowledge about the user. Section 6 will discuss techniques for acquiring some of this knowledge, and the use of this knowledge to enhance to determine the familiarity criteria of understandability is illustrated with examples in section 7.

5 The Need for User Models in Generating Explanations

Having explored the nature of explanation and good explanation, this section will argue that user models are needed to produce good explanations, and indicate how knowledge about the user can

[5] Ringle and Bruce [30] draw this distinction in their discussion of conversation failure. They call a failure to comprehend what is said an *input failure*, while a failure to assimulate the intended meaning of the statement is a *model failure*.

Preprint of: Robert Kass and Tim Finin, The need for user models in generating expert system explanation. International Journal of Expert Systems, v1, n4, pp 345-375, October 1988.

be used to improve explanations according to the criteria presented in section 4. First, however, a definition for the term "user model" is needed.

5.1 User Models

Providing a precise definition of what constitutes a user model is not easy to do. As a basis for discussion, we will adopt a definition proposed by Wahlster and Kobsa [38] that states:

> A *user model* is a knowledge source in a natural-language dialog system which contains explicit assumptions on all aspects of the user that may be relevant for the dialog behavior of the system.

The knowledge a user model keeps about a user may be quite varied, including assumptions about the user's goals, plans, preferences and attitudes, capabilities, and knowledge or beliefs [18]. User models also vary in their generality: a system might maintain *individualized* models for every user it encounters, with specific assumptions about each user's goals, preferences, beliefs, and so on; or the system might keep a *generic* user model that it applies to all users of the system. Generic user models often are not explicitly represented, but implicit in the design of the system as a whole. In fact, any system that interacts with the user can be said to have an implicit user model, if only by virtue of the assumptions the system builder made about the user while designing the system.

5.2 Why User Models are Needed

The discussion in sections 3 and 4 has provided a basis for understanding the role of user models in generating explanations. User models are not needed to generate explanations *per se*, since explaining is simply an illocutionary act—it is possible to provide an explanation without considering the hearer at all. However, to produce a good explanation the speaker must consider the likely perlocutionary effect of the explanation on the hearer: will the explanation be relevant to the hearer's goals?, is it likely to convince him of the point being justified?, and is he likely to understand it? To answer these questions, the speaker must reason about the hearer beliefs, goals, plans, and preferences.

Still, an expert system explanation component may not require an explicit user model to produce good explanations. If the anticipated class of system users is homogeneous in the beliefs and intentions of its members, an explanation component can be designed to produce good explanations for this class of individuals.

Frequently, though, the anticipated system users will vary in their knowledge and goals. For example, intelligent help systems, intelligent tutoring systems, or domains such as investment advising will have users who vary greatly in their knowledge of the system domain. In this case, to produce good explanations for all users, the explanation component will need to tailor the explanations it produces—based on the model it has of the user's beliefs, goals, and plans. Thus, to produce good explanations for a range of users requires an explanation component to make use of explicit user models.

To produce a relevant explanation, the explanation component must have knowledge of the user's goals and preferences. Not only must the system know the user's immediate goals, but also his higher goals and preferences, and his intended plan for accomplishing those goals. A relevant explanation may need to provide information to help a user achieve a higher goal or goals, or to correct the user's plan when it is faulty. Much work has been done in this area to identify what information about the user's plans and goals are needed, and how that information can be used to produce cooperative responses relevant to the user's situation [2, 27, 6].

Preprint of: Robert Kass and Tim Finin, The need for user models in generating expert system explanation. International Journal of Expert Systems, v1, n4, pp 345-375, October 1988.

The convincingness of a justification depends on the soundness of the argument presented, and the likelihood that the user will accept the justification. The soundness of the argument itself is independent of the user, but whether he will accept the argument is another matter. To produce a convincing justification, then, the system must consider whether the user believes the premises of the justification, and whether he considers the inferences made (or even the form of argument) valid. Thus, a robust model of the user's beliefs will be needed to help generate acceptable justifications.

Producing an explanation that the user will find understandable also requires knowledge of the user's beliefs about the domain. Here the user model is needed primarily to determine the familiarity of the user with the system domain. In this case, the explanation component needs to know the concepts and properties the user knows about, the terms he understands, and the relationships between entities, such as the relationship between reasoning steps that the user performs.

A deeper model of the user may also be required to estimate the user's ability to process the information communicated in the explanation. A *psychological* model of the user's ability to make inferences from the system's statements, or to fill in omitted reasoning steps can be useful in determining how to organize an explanation and in deciding how much information to include explicitly.

In summary, individualized models of the user are important, even necessary, to generate good explanations when system users vary in their goals and domain knowledge. In the remainder of this paper we will illustrate how a user model may be used when tailoring explanations in an investment advisory system. Section 6 presents work we have done on acquiring models of users' beliefs. Section 7 illustrates how such models can be acquired from a user-system dialog, and how the information about user beliefs acquired in this way enables an explanation component to tailor its explanations to individual users so they are more understandable.

6 Acquiring User Beliefs for Explaining

Using a detailed model of users' beliefs to support the generation of expert system explanations has previously appeared impractical, due to the difficulty in acquiring such a model. Techniques that emphasize explicit pre-encoding of user models (such as stereotypes [29]) are tedious, error-prone, and may potentially require more time to build than the domain knowledge base itself, due to the number of separate models necessary. On the other hand, acquiring the information from the user, either explicitly or implicitly, has not seemed feasible. Explicitly asking the user about his knowledge of the domain (as in the UMFE system [33]) can be very time consuming and potentially fraught with error due to the user's own misconceptions about what he knows, while implicit acquisition techniques have been viewed as either too slow to build a robust model or too unreliable in the conclusions they make.

Our current research [16, 19] indicates that this user model acquisition bottleneck can be overcome. The solution centers on a set of implicit acquisition rules that make reasonable inferences about a user's beliefs, based on the interaction between the system and the user, the system's knowledge base, and the existing model of the user. These rules were developed after study of an extensive collection of transcripts of conversations between advice-seekers and a human expert,[6] and have been implemented in the General User Model Acquisition Component, or GUMAC.

[6] The transcripts were made by Martha Pollack and Julia Hirschberg from the radio talk show "Harry Gross: Speaking about Your Money" broadcast on station WCAU in Philadelphia, February 1–5, 1982.

Preprint of: Robert Kass and Tim Finin, The need for user models in generating expert system explanation. International Journal of Expert Systems, v1, n4, pp 345-375, October 1988.

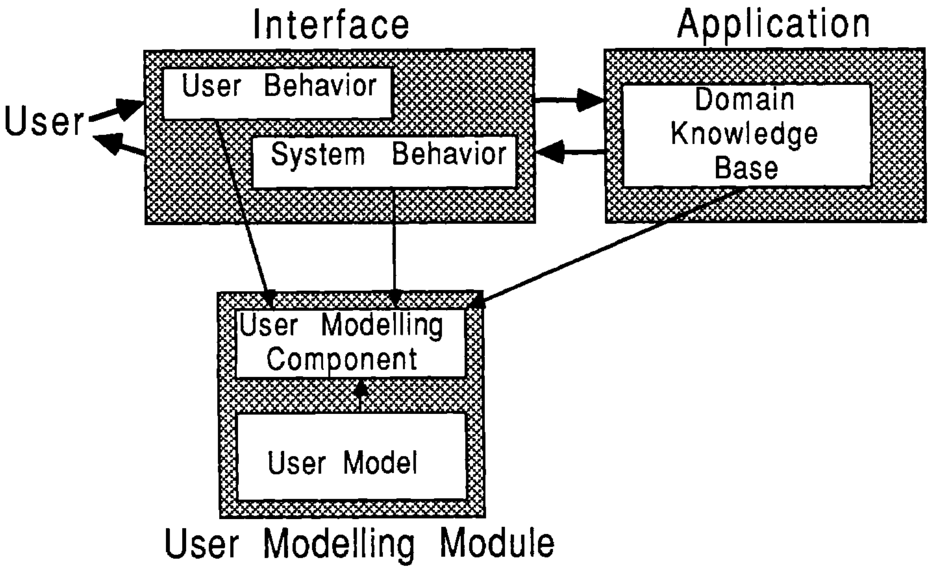

Figure 1: Sources of Knowledge for Implicit User Model Acquisition

6.1 The GUMAC System

GUMAC works in a cooperative advisory situation, where a user comes to the system seeking advice about a problem, and both the user and the system cooperate in solving that problem. Figure 1 illustrates the role GUMAC plays in an interactive system. The user interacts with an underlying application (the advisory expert system) through a user interface. The user modelling module (of which GUMAC is the model acquisition component) has access to the interface and the domain knowledge base and uses these to build the user model. GUMAC has been implemented in the context of an investment advising system, but the acquisition techniques are not limited to this domain. In fact, the rules make no assumptions about the domain knowledge, but instead depend on the type of interaction, i.e. a cooperative advisory interaction. The domain independence of these rules lends support to the feasibility of building a general user modelling system, as proposed by Finin [12].

The interface is assumed to have a natural language parser and semantic interpreter that produce an intermediate *meaning representation language* (MRL). In our implementation this component has been simulated by hand-translating English sentences into a LUNAR-style MRL [42].[7] GUMAC uses this intermediate representation as the basis for its reasoning about statements made by both the user and the system.

The user models built by GUMAC can be viewed as individual knowledge bases containing each user's beliefs. These user models may be queried by other components of the system, such as the application or the interface, to obtain yes/no answers about individual user's beliefs.[8] GUMAC

[7] We are making no special assumptions about the capability of the parser and semantic interpreter beyond the capabilities of current systems. The parser and semantic interpreter are not actually present in our test system because of the implementation time required.

[8] This is the simplest use for the user model. More sophisticated uses for user models, such as using the model to simulate the user, are possible and have been explored (See, for example, Wahlster and Kobsa's discussion of anticipation feedback loops in generating anaphoric responses [38].) Likewise, the responses given to queries about user beliefs can be more sophisticated as well, providing degree of belief measures or justifications for the beliefs held in the user model, instead of a simple yes/no answer.

Preprint of: Robert Kass and Tim Finin, The need for user models in generating expert system explanation. International Journal of Expert Systems, v1, n4, pp 345-375, October 1988.

```
(DEFCONCEPT stock
    (SPECIALIZES equity)
    (SPECIALIZES market-traded-security)
    (SPECIALIZES corporate-security)
    (ROLE market (VR stock-market))
    (ROLE own (VR corporation))
    (ROLE issuer (VR corporation))
    (ROLE par-value (VR dollar-amount))
    (ROLE dividend (VR quant-val-measure)))
```

Figure 2: A NIKL definition for the concept of stock. The "specializes" clauses express the fact that a stock is a kind of equity, market-traded-security, and a corporate-security; anything true of these concepts is true of stock as well. "VR" stands for the value restriction of a role.

is designed to acquire knowledge of what the user knows about the system's domain knowledge. Thus, the user models built by GUMAC can be thought of as *overlay models* [7], in that entities (such as concepts, properties, or actions) in the user model will always be a subset of those in the domain model of the application system. However, the user models GUMAC builds can represent different relations between these entities, so the user models are not strictly subsets of the domain model.

The domain knowledge is represented using EES, the Explanable Expert System [24], and consists of two types: definitional knowledge about the entities in the domain, and strategic (or reasoning) knowledge about how to solve problems in the domain. The definitional knowledge is represented in NIKL [23], a semantic network similar to the taxonomic component of KL-ONE [4]. NIKL has two types of entities: concepts (such as stock or equity), which can be expressed as 1-place predicates in a first-order logic, and roles representing relations between concepts (such as owner or interest-rate), which can be expressed as two place predicates. An example of a NIKL definition for the stock concept is illustrated in figure 2. GUMAC makes user model assertions about the user's knowledge of concepts, roles, whether roles apply to particular concepts, and the specialization relations between concepts.

The strategic knowledge in EES is represented using a goal and plan hierarchy. A plan consists of a capability description of the action it can accomplish and a method consisting of a sequence of steps to be performed to accomplish the capability description. To build an expert system in EES, the system designer writes a set of plans, provides a top level goal to be accomplished, then invokes an automatic program writer. The program writer examines the top level goal, finds plans capable of accomplishing this goal, selects one, instantiates it, then posts the steps in the plans's method as subgoals and recursively tries to find plans to accomplish them. The result of this process is a goal refinement structure representing the system's reasoning method to achieve the top level goal. GUMAC makes assertions to the user model about the user's beliefs about the goal-subgoal relations between actions in this hierarchy and the user's knowledge of the actions themselves.

6.2 The Acquisition Rules

Implicit acquisition of user models is feasible because of the structure of human communication and reasoning. The fact that a particular type of conversation is being held provides constraints

Preprint of: Robert Kass and Tim Finin, The need for user models in generating expert system explanation. International Journal of Expert Systems, v1, n4, pp 345-375, October 1988.

on the expected behavior of the conversation participants that can serve as a basis for inferring an individual's beliefs during the conversation. Likewise, the fact that an individual is human creates expectations about how he reasons and, thus, the beliefs he holds.

The acquisition rules are *reasonable* inference rules, but they can make mistakes. The goal is not to produce certain knowledge about the user, but rather the assumptions that a reasonable conversational participant would make about the user in the same situation. Thus, the acquisition rules can be viewed as default rules in a default logic [28]: they draw conclusions that are reasonable to believe unless information to the contrary is encountered.

One important set of user model acquistion rules are the communication rules, based on Grice's maxims for cooperative communication [14]. Grice has proposed these maxims as a way of describing the behavior of cooperative conversational participants. By assuming that users are cooperating with the system,[9] the communication rules make inferences about user beliefs based on these maxims.

Direct Statement Rule The *direct statement* rule is based on the maxim of quality ("Only say that which you believe to be true"), and can be expressed as:

$$\text{coop-agent(User)} \wedge \text{tell(User, System, P)} \longrightarrow \text{believe(User, P)}.$$

Here, P will be a logical form expressing the content of the user's statement in the intermediate MRL. In itself this is not too useful, so the expression is decomposed into a set of assertions about the user's beliefs about domain concepts, roles, and their relations. Figure 3 illustrates a sample user statement and the associated MRL for that statement. The "saying" portion of this MRL expression will be asserted as a user belief, along with a list of assertions about the user's knowledge of domain entities, some of which are illustrated in figure 4. Note that an assertion such as "bel(U, concept(T-Bill))" does not mean the user knows all about T-Bills, only that he has some knowledge of the concept. Other assertions, such as "bel(User, role(T-Bill, interest-rate))," are necessary to indicate the aspects of U's knowledge of T-Bills.

Relevancy Rule The *relevancy rule*, based on Grice's maxim of relation, draws conclusions about the user's beliefs about the strategic knowledge of the domain. This rule presumes that since the user is cooperating, his contributions will be relevant to the current conversational goal. In terms of the EES representation, this means the user believes his action is a step (or subgoal) in achieving the system's current goal. The relevancy rule can be expressed as follows:

$$\begin{gathered}\text{coop-agent(User)} \wedge \text{tell(User, System, P)} \wedge \text{current-goal(System, G)} \longrightarrow \\ \text{bel(User, subgoal(tell(User, System, P), G))}.\end{gathered}$$

In practice, the system's current goal is easy to determine from the goal refinement structure produced by the EES automatic program writer. Furthermore, this goal can be assumed to be a mutual belief held by the user and system, since the system will explicitly state its goal by asking a question, such as "what is your yearly income?"[10] The relevancy rule provides an example of how the acquisition rules can make conclusions that are not strictly a subset of the system's domain model. For example, if the current goal is to determine the user's yearly income, and the user provides information about his property tax payments, the relevancy rule will conclude that the user believes property tax information is needed to determine income, when the system knows it is not.

[9] This is a reasonable assumption, since the user has come to the system to receive advice.

[10] In fact, another acquisition rule concludes that if the system asks a question, then the user believes that the goal associated with that question is the current goal.

Preprint of: Robert Kass and Tim Finin, The need for user models in generating expert system explanation. International Journal of Expert Systems, v1, n4, pp 345-375, October 1988.

User: "I have a $10,000 T-Bill at 7-1/2% interest."

```
(FOR THE t1 / tell
        : (speaker t1 User)
        : (addressee t1 System)
        : (saying t1
                  (FOR THE tb1 / treasury-bill
                          : (interest-rate tb1
                                  (FOR THE pct1 / percentage
                                          : (measurement-unit pct1 number)
                                          : (value pct1 7.5)))
                          : (face-value tb1
                                  (FOR THE d1 / dollar-amount
                                          : (measurement-unit d1 dollar)
                                          : (value d1 10000)))
        ; (owned-by tb1 User)))))
```

Figure 3: A User Statement and Associated MRL

bel(User, concept(interest-rate)
bel(User, concept(face-value)
bel(User, concept(interest-rate-domain)
bel(User, concept(face-value-domain)
bel(User, concept(percentage)
bel(User, concept(dollar)
bel(User, concept(treasury-bill)
bel(User, concept(number)
role(treasury-bill, interest-rate)
role(treasury-bill, face-value)
role(interest-rate-domain, interest-rate)
role(face-value-domain, face-value)
bel(User, concept(interest-rate-range)
bel(User, concept(face-value-range)
bel(User, isa(treasury-bill, interest-rate-domain)
bel(User, isa(treasury-bill, face-value-domain)

Figure 4: Assertions Made by the Direct Statement Rule

Preprint of: Robert Kass and Tim Finin, The need for user models in generating expert system explanation. International Journal of Expert Systems, v1, n4, pp 345-375, October 1988.

Sufficiency Rule A third communication rule, the *sufficiency rule*, is based on the maxim of quantity, which states "make your contribution as informative as necessary, but not more so." The sufficiency rule deals with the situation where the system knows that a certain action must be performed by the user to accomplish the current goal, but the user fails to perform that action. In this case, there are several possibilities: (1) the user doesn't know that current goal, (2) the user doesn't believe the action is relevant to the goal, (3) the user believes the system can achieve the goal without the action being performed, or (4) the user does not believe he can perform the action. This can expressed as follows:

$$\begin{array}{c}
\text{coop-agent(User)} \wedge \text{current-goal(System, G1)} \wedge \text{subgoal(G2, G1)} \wedge \neg\text{do(User, G2)} \longrightarrow \\
\neg\text{bel(User, current-goal(System, G1))} \vee \neg\text{bel(User, subgoal(G2, G1))} \vee \\
\text{bel(User, achieve(System, G1))} \vee \text{bel(User, }\neg\text{achieve(User, G2)).}
\end{array}$$

Other Rules Other implicit acquisition rules reason on the knowledge base and the current user model. Such rules include transitivity rules for goals and concepts, the inheritance rule, and a generalization rule for concepts:

$$\begin{array}{llllll}
\text{bel(User, isa(A, B))} & \wedge & \text{bel(User, isa(B, C))} & \longrightarrow & \text{bel(User, isa(A, C))} & \\
\text{bel(User, subgoal(G1, G2))} & \wedge & \text{bel(User, subgoal(G2, G3))} & \longrightarrow & \text{bel(User, subgoal(G1, G3))} & \\
\text{bel(User, isa(A, B))} & \wedge & \text{bel(User, role(B, R))} & \longrightarrow & \text{bel(User, role(A, R))} & \\
\text{bel(User, concept(A)} & \wedge & \text{bel(User, concept(B))} & \wedge & \text{bel(User, concept(C))} & \wedge \\
 & & \text{bel(System, concept(D))} & \wedge & \text{bel(System, isa(A, D))} & \wedge \\
 & & \text{bel(System, isa(B, D))} & \wedge & \text{bel(System, isa(C, D))} & \longrightarrow \\
 & & \text{bel(User, concept(D)} & \wedge & \text{bel(User, isa(A, D))} & \wedge \\
 & & \text{bel(User, isa(B, D))} & \wedge & \text{bel(User, isa(C, D)).} &
\end{array}$$

A similar generalization rule exists for goals, but is difficult to express in a simple logical notation. Essentially, if the user knows all the plan steps where he is an agent, then the goal generalization rule will conclude that the user knows the goal the plan is capable of achieving.

During the course of a user-system dialog, these user model acquisition rules make a large number of assertions about the user's beliefs about the system's domain. At the end of this dialog, GUMAC has built a robust model of the user's domain knowledge with respect to the topics discussed in the dialog. This model will contain few assertions of user's beliefs beyond the discourse topics of the dialog, but the model built is sufficient to support the tailoring of explanations to individual users. Examples of how a model is built by the acquisition rules and used to tailor explanations are presented in the next section.

7 An Extended Example

This section illustrates how a user model may be acquired from an on-going dialogue, and how that model can be used to tailor explanations so they are more understandable. To this end, the interaction of two users with an investment advisory system is presented. Each user has the same goal for using the system, and is in the same financial situation, so the system's recommendation will be the same in each case, and for the same reasons. The users differ, however, in their knowledge about the investment domain. These differences are evident in the way they interact with the system, causing the acquisition rules to construct different models for each user, and suggesting that the explanations given to each user should differ.

Preprint of: Robert Kass and Tim Finin, The need for user models in generating expert system explanation. International Journal of Expert Systems, v1, n4, pp 345-375, October 1988.

```
Goal:    recommend(System, User, invest(User, Security))
Method:  1) determine(System, acceptable-risk(User)) ⟶ AR
         2) determine(System, required-income(User)) ⟶ RI
         3) determine(System, required-liquidity(User)) ⟶ RL
         4) choose-security(System, AR, RI, RL) ⟶ Sec
         5) tell(System, User, invest(User, Sec))
```

Figure 5: Recommend Investment Plan

In this scenario, each user wants to know how to invest $10,000 he now has in a savings account. The top level goal of the investment advisor (represented in EES) is for the system to recommend to the user that he invest in a particular security, accomplished by the RECOMMEND-INVESTMENT plan, illustrated in figure 5. Much of the user-system interaction focuses on determining the user's income, obligations, and current investments in order to determine the user's income and liquidity requirements, and the degree of risk he can accept in his investments. The system then compares these constraints with its knowledge of potential investment securities to select the most appropriate security to recommend.

A Knowledgeable User

The first dialogue proceeds as follows:

(1) U1: Could you recommend how I should invest $10,000 I have in my savings account?

(2) S: What are your current investments?

(3) U1: I have a $40,000 6-month CD at 8% interest,
a $10,000 T-Bill at $7\frac{1}{2}$% interest,
200 shares of AT&T which I bought at 32,
$14,000 in a savings account,
and $1000 in checking.

(4) S: What is your yearly income?

(5) U1: I earn $35,000 and my wife makes $15,000, plus we get $5000 from our investments.

(6) S: What obligations do you have?

(7) U1: I pay $300 on a car loan and $900 on my home loan.

(8) I have no dependents.

At this point the system has sufficient information to make the recommendation that the user should invest in growth stocks.[11] Furthermore, the model acquisition rules have built a substantial model of the user's beliefs about investing, portions of which are illustrated in figures 6 and 7.

[11] The investment advisory system we are using has limited domain knowledge, so the interaction stops at this point. A more robust system would probably require a longer dialogue in order to gather the necessary information to make a decision. We feel this would enhance the effectiveness of the user model acquisition component, since it would have more information to work from.

Preprint of: Robert Kass and Tim Finin, The need for user models in generating expert system explanation. International Journal of Expert Systems, v1, n4, pp 345-375, October 1988.

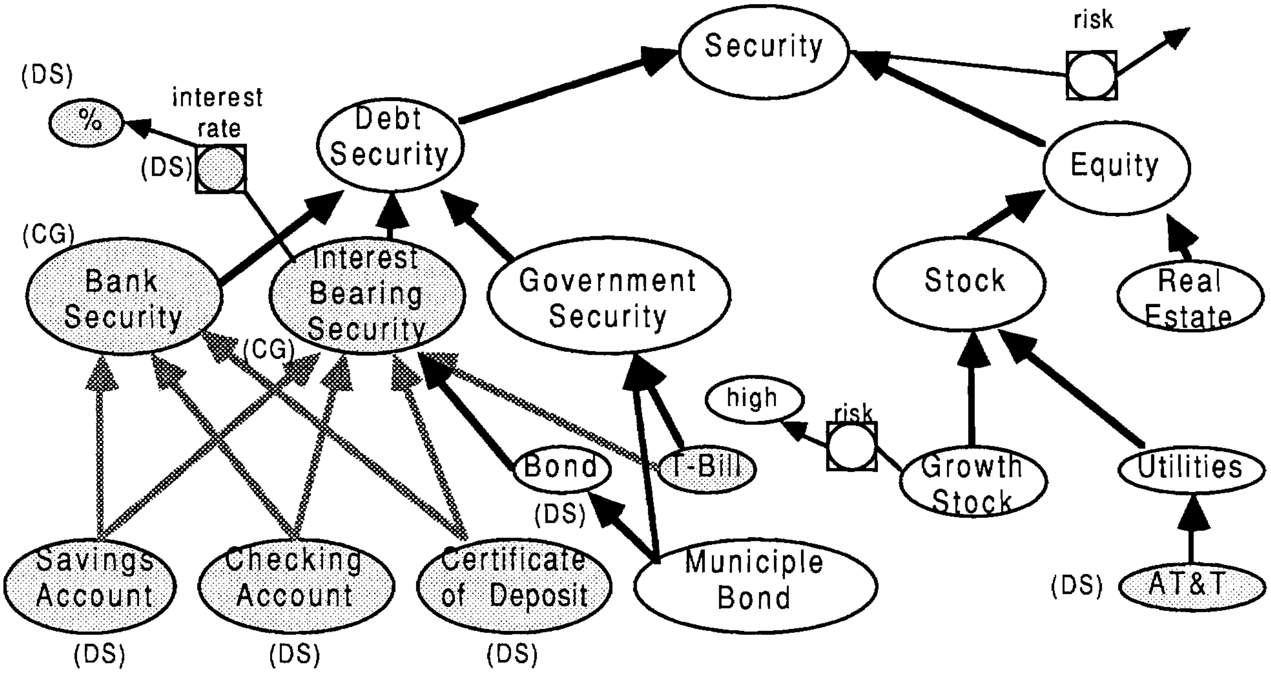

Figure 6: A Model of U1's Knowledge of Investment Securities
Concepts and Roles U1 is believed to know are shown in gray, with an associated rule indicating why the user is believed to know it. "DS" stands for the direct statement rule, while "CG" indicates the concept generalization rule was used.

Preprint of: Robert Kass and Tim Finin, The need for user models in generating expert system explanation. International Journal of Expert Systems, v1, n4, pp 345-375, October 1988.

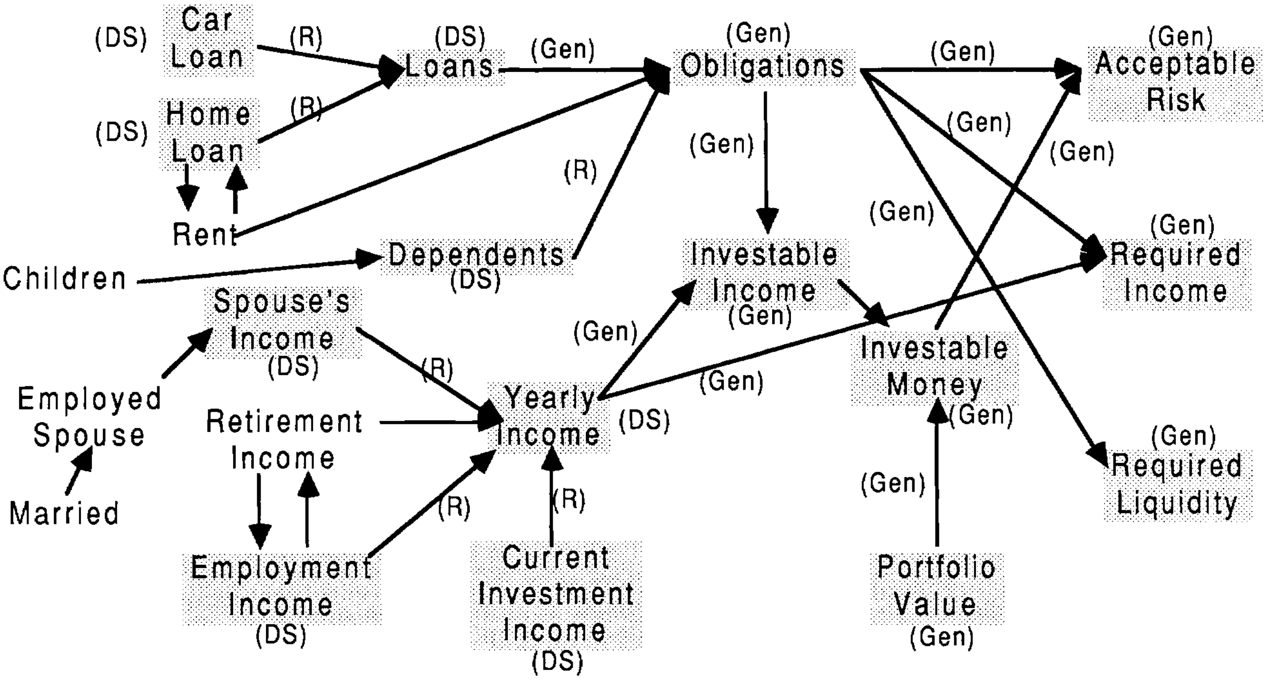

Figure 7: A Model of U1's Knowledge of Property Dependencies
Properties U1 is believed to know are shown in gray. Associated with each property or relation U1 is believed to know is the rule that asserted the conclusion. "DS" stands for the direct statement rule, "R" for the relevancy rule, and "Gen" for the goal generalization rule.

Preprint of: Robert Kass and Tim Finin, The need for user models in generating expert system explanation. International Journal of Expert Systems, v1, n4, pp 345-375, October 1988.

```
Goal:    determine(System, yearly-income(User))
Method:  1) askref(System, User, yearly-income(User))
         2) determine(System, employment-income(User)) ⟶ EI
         3) determine(System, retirement-income(User)) ⟶ RI
         4) determine(System, spouse's-income(User)) ⟶ SI
         5) determine(System, investment-income(User)) ⟶ II
         6) add(System, EI, RI, SI, II) ⟶ YI
```

Figure 8: Determine Yearly Income Plan

Figure 6 depicts a small portion of the definitional knowledge represented in the NIKL semantic network. In answering the system's questions U1 has made several direct statements about his financial situation. From these statements the direct statement infers that U1 believes what he has said, and thus knows about concepts such as SAVINGS-ACCOUNT and INTEREST-RATE. Building on these assertions, the concept generalization rule can infer U1's knowledge of more abstract concepts, such as inferring U1's knowledge of BANK-SECURITIES from his knowledge of SAVINGS-ACCOUNT, CHECKING-ACCOUNT, and CERTIFICATE-OF-DEPOSIT. Furthermore, the transitivity and inheritance rules have made a large number of inferences about concept properties and subsumption relations that are not explicitly depicted in the figure.

Figure 7 illustrates a portion of the investment advisor's reasoning structure. In this domain, the system can be viewed as performing a series of actions to determine information. This figure is a graph showing some of the pieces of information the system must determine. For example, to determine the user's yearly income, the system needs to determine the user's employment and retirement income, his spouse's income, and his income from current investments. In EES, this is represented as the six-step plan illustrated in figure 8. Only the plan steps that involve user action are illustrated in figure 7. In this dialogue, when U1 was asked about his yearly income and obligations, he provided a large amount of information that satisfied the subgoal requirements of the DETERMINE-YEARLY-INCOME and DETERMINE-OBLIGATIONS plans. The relevancy rule has made assertions reflecting this, since U1's action of telling the system this information (such as telling the system his investment income) achieves the "determine" goals (such as DETERMINE-INVESTMENT-INCOME). These assertions indicate that the user knows how yearly-income and obligations are determined. Furthermore, this knowledge enables the goal generalization rule to reason further that U1 knows how REQUIRED-INCOME, ACCEPTABLE-RISK, and so on are determined.

After this dialogue, the system makes its recommendation "You should invest in growth stocks," to which U1 may ask "Why?" This question may be answered in many ways, but for this example we will assume that a standard type of expert system explanation is given—a description of the steps the system took to reach this conclusion. The user model that has been acquired is now useful in applying the familiarity criteria to determine what information to include in the explanation. To be successful, an explanation should be grounded in the user's own knowledge of the domain, a requirement that affects how deeply the explanation must delve into the system's goal refinement structure before reaching actions and terms the user knows. In fact, if the explanation component assumes the user has no knowledge of the domain beyond that directly evidenced in his statements, the explanation may include the majority of this refinement structure.

From the dialogue the relevancy and goal generalization rules have inferred that the user knows a lot about the system's reasoning, so a large portion of the refinement structure can be omitted

Preprint of: Robert Kass and Tim Finin, The need for user models in generating expert system explanation. International Journal of Expert Systems, v1, n4, pp 345-375, October 1988.

from the explanation. In fact, starting from the top level RECOMMEND-INVESTMENT plan presented in figure 5, the system does not need to explain how it determined REQUIRED-INCOME, REQUIRED-LIQUIDITY, and ACCEPTABLE-RISK because the user model indicates that U1 knows this already. Thus, the explanation only needs to concentrate on the CHOOSE-SECURITY step. Following is an example of the type of explanation the system could produce for U1:

> S: To recommend an investment, I choose a security that maximizes return on investment, while satisfying your income, liquidity, and risk needs. Since you can accept high risk, debt securities (such as bank securities or government-issued securities like T-Bills) are ruled out, so you should invest in an equity. Real estate is not appreciating well presently, so stock would be a good investment. Since you can accept a fair amount of risk, I recommend stock with a high potential for return, such as stock in growth-oriented companies.

In addition to deciding how much explanation to give, the system must consider how to express that information; the user model can help here as well. Notice that in the explanation above the term "debt security" is introduced with examples of types of debt securities. The user model contains no assertions that the user knows what a debt security is, so the explanation generator tries to introduce the term with respect to concepts that the user model indicates he does know about, such as bank securities. That is not sufficient, so it also says that government securities are types of debt securities, giving a partial definition for this term and an example of a government security that the user knows about.

Although an explanation generation component is not included in the GUMAC implementation, the capabilities assumed are not unreasonable. Weiner's BLAH system [41] used a similar approach to generate explanations, using a user model to decide what reasoning steps to include. Moore and Paris [22] are currently implementing a plan-based explanation generator for the EES system that reasons about the user's knowledge state to control the generation process. It should require little work to integrate GUMAC with their system.

A Novice User

The dialogue with a novice investor is significantly different. Where the knowledgeable user volunteered relevant information when a question was asked, the novice does not know what information is relevant (or does not know the information at all), hence, the system must ask further questions to elicit this information from him. For example, where U1 volunteered information about his wife's income and dependents, in the following conversation the system must ask several questions to get this information from U2.

(9) S: What is your yearly income?

(10) U2: I earn $35,000.

(11) S: Are you married?

(12) U2: Yes.

(13) S: Does your spouse work?

(14) U2: Yes.

(15) S: What is her income?

(16) U2: She makes $15,000.

(17) S: Do you have any dependents?

Preprint of: Robert Kass and Tim Finin, The need for user models in generating expert system explanation, International Journal of Expert Systems, v1, n4, pp 345-375, October 1988.

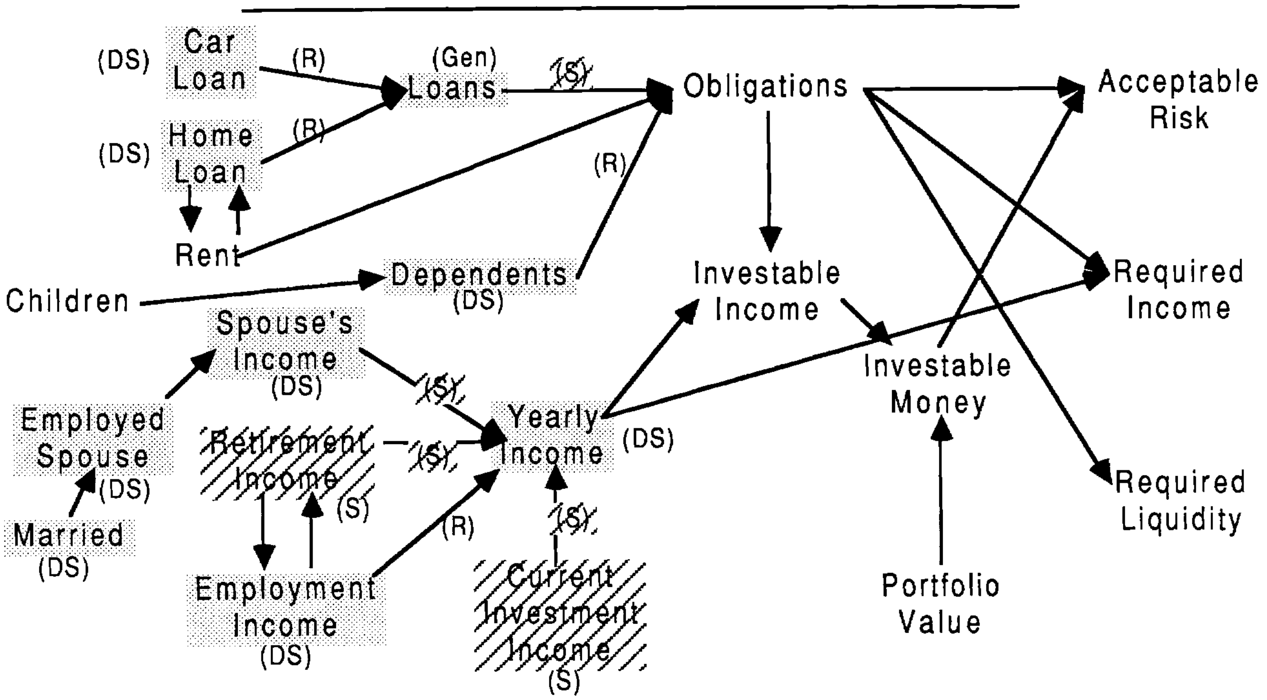

Figure 9: A Model of U2's Knowledge of Property Dependencies
Properties U2 is believed to know are shown in gray. Properties and relations U2 is believed to not know are shown with hatching. Associated with each property or relation U2 is believed to know is the rule that asserted the conclusion. "DS" stands for the direct statement rule, "R" for the relevancy rule, and "S" for the sufficiency rule.

(18) U2: No.

(19) S: Are you making any loan payments?

(20) U2: I pay $300 on a car loan and $900 on my home loan.

The portion of the user model built for U2, illustrated in figure 9, together with the comparable model for U1 in figure 7 indicates the significant differences between the models built for each user.

An explanation for U2 will require much more detail. To explain the rationale for its recommendation, the system will need to give a detailed account of how it determined the user's income, liquidity, and risk requirements, in addition to explaining how it chose to recommend growth stocks. In fact, the explanation would be most effective if given in pieces, so that the system could check that the user understands what has been said at each step. An example of one portion of an explanation for U2, concerning his income needs, is illustrated below.

S: Since you don't have substantial loans, and you have no dependents, you don't have many financial obligations to meet (things you will need to pay for). Furthermore, your yearly income from your job and your wife's, and from your current investments is more than sufficient to meet those obligations, so your future investments don't need to produce regular income (such as interest payments).

Here, to explain how REQUIRED-INCOME was determined, the system first explains how OBLIGATIONS and YEARLY-INCOME were determined, grounding the explanation in the simple steps U2 knows about.

Preprint of: Robert Kass and Tim Finin, The need for user models in generating expert system explanation. International Journal of Expert Systems, v1, n4, pp 345-375, October 1988.

Discussion

A model of a user's knowledge and beliefs can significantly affect the explanations generated by an expert system. In the previous example, users at two extremes of knowledge about the domain were chosen to illustrate the types of explanations that might be generated if knowledge about the user is available. How does this compare to systems that have a fixed method for generating explanations (or a fixed set of assumptions about the user)? Such systems always generate the same explanation regardless of the user. If the system assumed that all users are knowledgeable about investments, then U2 would receive an explanation similar to that generated for U1 in the example. Such a terse explanation, using terms U2 did not understand and omitting reasoning steps necessary for him to follow the justification would not be understandable.

On the other hand, if the system assumed all users were novices, then U1 would receive an explanation similar to that given to U2. In this case, U1 would probably understand the explanation (although the length of the explanation might cause U1 to skip the explanation or only read portions of it), but the same understanding could be achieved with a much simpler explanation. In this case, the explanation given U1 fails to be good because it is too long and complex.

In most situations, actual users will fall somewhere between the extremes used in this example (although the descriptions of the knowledge of U1 and U2 are reasonable). Yet, even if the system chooses a generic model for some "average" user, if there is significant variability among users (which is the case for many domains, such as investment advising), then explanations based on this generic model of the user will frequently fail to be as good as they could be. Thus, a model of the user's beliefs is an important component to generating good explanations, and the implicit acquisition technique we have presented makes it practical to include such models in an expert system.

8 Conclusions

An explanation facility is an important component of an expert system, perhaps the most important component. Thus, the quality of a system's explanations will significantly affect its acceptability and effectiveness. This paper has explored the nature of explanation, the components of a good explanation, and argued that frequently systems will need to tailor explanations to individual users and thus need a model of those users' beliefs. Furthermore, it is practical to acquire such user model implicitly, a method that greatly reduces the effort required to incorporate user models into explanation systems.

The user model acquisition method described in this paper has been purposely restricted to only implicit techniques to demonstrate the power and feasibility of such an approach. This approach is probably too extreme for practical systems, however. In the GUMAC system, no explicitly acquired information is kept about the user—when a new user is encountered, the model describing his beliefs is a blank slate. For a practical system a combination of implicit and explicit acquisition would be more effective. If the user model is to be used to help determine system behavior during the dialogue, an initial, explicitly acquired model would be useful to support the first portions of the conversation. Then, as the dialogue proceeds, the implicit acquisition rules will progressively refine this initial model to correspond to the specific user. Such a method would still minimize the explicit acquisition required, since the initial model could be a generic model describing the "average" system user.[12]

[12] See [17] a discussion of how stereotypic user models can be integrated with implicit acquisition methods in a general user modelling system.

Preprint of: Robert Kass and Tim Finin, The need for user models in generating expert system explanation. International Journal of Expert Systems, v1, n4, pp 345-375, October 1988.

This work suggests several possible directions for studying the role of user modelling in expert system explanation. First, the classification of good explanation given in section 4 is just an initial attempt to characterize an area that is not well understood. More work is needed to discover what makes an explanation good—this work will contribute greatly to the effectiveness of future expert systems. A study of the types of knowledge about the user needed for explanation generation is a second area for further research. Section 5 provided a general catagorization of these types, but to actually generate good explanations, these categories must be defined in detail. A related research area is to study how an explanation facility uses a user model. This study could lead to an understanding of what general services a user model should be expected to provide to the system components that use it. A fourth, very important need is to develop a measure or measures of the quality of an explanation. Currently, two explanations cannot be objectively compared to determine which is better. Finally, the work on acquiring a user model described in section 6 is quite limited in the types of information it can acquire, and the types of interactions from which it can acquire that information. To extend the explanation capabilities of an expert system, user model acquisition capabilities must be extended as well.

Expert system explanation and user modelling are both relatively new fields of study. This paper has demonstrated the importance of user modelling for explanation, and provides examples of how to acquire and use such information to improve the understandability of explanations. Furthermore, this work suggests many areas for further research, research that should result in substantial improvements in the explanations produced by expert systems.

Preprint of: Robert Kass and Tim Finin, The need for user models in generating expert system explanation. International Journal of Expert Systems, v1, n4, pp 345-375, October 1988.

References

[1] Peter Achinstein. *The Nature of Explanation.* Oxford University Press, Oxford, 1983.

[2] James F. Allen and C. Raymond Perrault. Analyzing intention in utterances. *Artificial Intelligence*, 15:143–178, 1980.

[3] J. Austin. *How to Do Things with Words.* Oxford, New York, 1965.

[4] R. J. Brachman and J. G. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9:171–216, 1985.

[5] Bruce G. Buchanan and E. H. Shortliffe. Human engineering of medical expert systems. In Bruce G. Buchanan and Edward H. Shortliffe, editors, *Rule-Based Expert Systems*, pages 599–612, Addison-Wesley, Reading, MA, 1984.

[6] Sandra Carberry. Modeling the user's plans and goals. *Computational Linguistics*, Special Issue on User Modelling, 1988.

[7] Brian Carr and Ira P. Goldstein. *Overlays: A Theory of Modelling for Computer Aided Instruction.* Technical Report A. I. Memo 406, MIT Artificial Intelligence Laboratory, Cambridge, Massachusetts, 1977.

[8] Daniel Chester. The translation of formal proofs into english. *Artificial Intelligence*, 7:261–278, 1976.

[9] William J. Clancey. The epistemology of a rule-based expert system — a framework for explanation. *Artificial Intelligence*, 20:215–251, 1983.

[10] William J. Clancey. Tutoring rules for guiding a case method dialogue. In D. Sleeman and J. S. Brown, editors, *Intelligent Tutoring Systems*, pages 201–226, Academic Press, New York, 1982.

[11] William J. Clancey and Reed Letsinger. NEOMYCIN: reconfiguring a rule-based expert system for application to teaching. In 7^{th} *International Conference on Artificial Intelligence*, pages 829–836, 1981.

[12] Tim Finin. GUMS—a general user modelling shell. In Alfred Kobsa and Wolfgang Wahlster, editors, *User Models in Dialog Systems*, Springer Verlag, Berlin—New York, 1988.

[13] Alan Garfinkel. *Forms of Explanation.* Yale University Press, New Haven, 1981.

[14] H. P. Grice. Logic and conversation. In P. Cole and J. L. Morgan, editors, *Syntax and Semantics*, Academic Press, New York, 1975.

[15] Carl G. Hempel. *Aspects of Scientific Explanation.* The Free Press, New York, 1965.

[16] Robert Kass. *Implicit Acquisition of User Models in Cooperative Advisory Systems.* Technical Report MS-CIS-87-05, Department of Computer and Information Science, University of Pennsylvania, 1987.

[17] Robert Kass and Tim Finin. A general user modelling facility. In *Proceedings of CHI'88*, 1988.

Preprint of: Robert Kass and Tim Finin, The need for user models in generating expert system explanation. International Journal of Expert Systems, v1, n4, pp 345-375, October 1988.

[18] Robert Kass and Tim Finin. Modelling the user in natural language systems. *Computational Linguistics*, Special Issue on User Modelling, 1988.

[19] Robert Kass and Tim Finin. Rules for the implicit acquisition of knowledge about the user. In *Proceedings of the 6th National Conference on Artificial Intelligence*, pages 295–300, 1987.

[20] K. R. McKeown. *Text Generation—Using Discourse Strategies and Focus Constraints to Generate Natural Language Text.* Cambridge University Press, 1985.

[21] Kathleen R. McKeown. Tailoring explanations for the user. In 9th *International Conference on Artificial Intelligence*, pages 794–798, 1985.

[22] Johanna D. Moore and Cecile L. Paris. *Constructing Coherent Text Using Rhetorical Relations.* Technical Report , USC/Information Sciences Institute, 1988. Submitted to 1988 Cognitive Science Conference.

[23] M. G. Moser. *An Overview of NIKL, The New Implementation of KL-ONE.* Technical Report 5421, Bolt, Beranek and Newman, 1983.

[24] Robert Neches, William R. Swartout, and J. Moore. Enhanced maintenance and explanation of expert systems through explicit models of their development. *IEEE Transactions on Software Engineering*, SE-11(11):1337–1351, 1985.

[25] Robert Neches, William R. Swartout, and J. Moore. Explanable (and maintainable) expert systems. In 9th *International Conference on Artificial Intelligence*, 1985.

[26] Cecile L. Paris. Tailoring object descriptions to a user's level of expertise. *Computational Linguistics*, Special Issue on User Modelling, 1988.

[27] Martha E. Pollack. *Inferring Domain Plans in Question-Answering.* PhD thesis, Department of Computer and Information Science, University of Pennsylvania, 1986.

[28] Raymond Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1):81–132, 1980.

[29] Elaine Rich. User modelling via stereotypes. *Cognitive Science*, 3:329–354, 1979.

[30] Martin H. Ringle and Bertram C. Bruce. Conversation failure. In Wendy G. Lehnert and Martin H. Ringle, editors, *Strategies for Natural Language Processing*, pages 203–221, Lawrence Erlbaum Associates, Hillsdale, NJ, 1980.

[31] Robert Rubinoff. Explaining concepts in expert systems: the CLEAR system. In *Proceedings of the Second Conference on Artificial Intelligence Applications*, pages 416–421, 1986.

[32] Wesley C. Salmon. *Scientific Explanation and the Causal Structure of the World.* Princeton University Press, Princeton, New Jersey, 1984.

[33] D. H. Sleeman. UMFE: a user modelling front end subsystem. *International Journal of Man-Machine Studies*, 23:71–88, 1985.

[34] William R. Swartout. XPLAIN: a system for creating and explaining expert consulting programs. *Artificial Intelligence*, 21:285–325, 1983.

Preprint of: Robert Kass and Tim Finin, The need for user models in generating expert system explanation. International Journal of Expert Systems, v1, n4, pp 345-375, October 1988.

[35] William R. Swartout and Stephen W. Smoliar. Explaining the link between causal reasoning and expert behavior. In *Proceedings of the Symposium on Computer Applications in Medical Care*, 1987. Also to appear in "Topics in Medical Artificial Intelligence"; Miller, P. L. (ed), Springer-Verlag.

[36] R. L. Teach and E. H. Shortliffe. An analysis of physician attitudes regarding computer-based clinical consultation systems. In Bruce G. Buchanan and Edward H. Shortliffe, editors, *Rule-Based Expert Systems*, pages 635–652, Addison-Wesley, Reading, MA, 1984.

[37] B. van Fraassen. *The Scientific Image.* Clarendon Press, Oxford, England, 1980.

[38] Wolfgang Wahlster and Alfred Kobsa. Dialog-based user models. *Proceedings of the IEEE*, 74(7), 1986.

[39] J. W. Wallis and Edward H. Shortliffe. Customizing explanations using causal knowledge. In Bruce G. Buchanan and Edward H. Shortliffe, editors, *Rule-Based Expert Systems*, Addison-Wesley, Reading, MA, 1984.

[40] Bonnie Webber and Aravind Joshi. *Taking the Initiative in Natural Language Data Base Interactions: Justifying Why.* Technical Report MS-CIS-82-1, Department of Computer and Information Science, University of Pennsylvania, 1982.

[41] J. L. Weiner. Blah, a system which explains its reasoning. *Artificial Intelligence*, 15:19–48, 1980.

[42] W. A. Woods. *Semantics and Quantification in Natural Language Question Answering.* Technical Report 3687, Bolt, Beranek and Newman, 1977.