Artificial Intelligence 54 (1992) 33–70
Elsevier

Reconstructive expert system explanation

Michael R. Wick

Computer Science Department, University of Wisconsin—Eau Claire, Eau Claire, WI 54702, USA

William B. Thompson

Department of Computer Science, University of Utah, Salt Lake City, UT 84112, USA

Received October 1989
Revised March 1991

Abstract

Wick, M.R. and W.B. Thompson, Reconstructive expert system explanation, Artificial Intelligence 54 (1992) 33–70.

Existing explanation facilities are typically far more appropriate for knowledge engineers engaged in system maintenance than for end-users of the system. This is because the explanation is little more than a trace of the detailed problem-solving steps. An alternative approach recognizes that an effective explanation often needs to substantially reorganize the actual line of reasoning and bring to bear additional information to support the result. Explanation itself becomes a complex problem-solving process that depends not only on the actual line of reasoning, but also on additional knowledge of the domain. This paper presents a new computational model of explanation and argues that it results in significant improvements over traditional approaches.

1. Introduction

A computer is generally poor at explaining its problem solving to a human user. Early work on expert systems suggested an explicit knowledge base of expert-defined, problem-solving rules might be used to explain the system's reasoning. During the past decade, research has been conducted on ways of using this knowledge base to explain the expert system's actions and conclusions. Although significant advances have been made, automatically generated explanations still suffer from several important flaws. The underlying premise of previous work is that the basis of the explanation is the trace of the expert system's *line of reasoning* [8]. Another approach is possible that for certain audiences will overcome many of the problems evident in earlier explanations.

A human expert, when asked to account for complex reasoning, rarely does so exclusively in terms of the process used to solve the problem [6]. Instead, an expert tends to reconstruct a 'story' that accounts for the problem solving. This story reflects the expert's *line of explanation* [20]—not necessarily the same as, or even a subset of, the original line of reasoning.

The idea of viewing automated explanation as requiring a new problem-solving session dates back to the SOPHIE system [1]. In SOPHIE, an analytic subsystem determined voltages in a circuit. Although the subsystem was very accurate in computing such voltages, its analytic nature precluded the ability to explain how those voltages were derived. This led to the addition of a rule-based subsystem that, when supplied with the voltages computed by the analytic subsystem, could rederive the voltage values and maintain a trace of the rules used as an explanation of how those voltages were derived.

In this paper, we describe a related approach in which a largely distinct knowledge-based explanation system is used to generate explanations for a separate knowledge-based problem-solving system. The advantage of this approach can be seen in the following example showing the line of reasoning taken by an inspector attempting to find the cause of the excessive load on a risk analysis (based on [9]).

> . . . the debris on top of the dam indicates a recent flood. The water markings on the abutments do too. I suspect the flood is the cause of the excessive load. No, the duration of the flood wasn't long enough. Sometimes settlement has these same features. Perhaps settlement is involved. That would account for the high uplift pressures indicated by the slow drainage over time. But the damage to the drainage pipes isn't right. It must be erosion causing the dam to settle more at the toe. Yes, erosion is causing the excessive load . . .

Note that the inspector is using a heuristic, data-driven problem-solving process. Key symptoms are extracted from the data and used to drive the process. After the evaluation is made, the field inspector is asked to explain the reasoning that led to the conclusion.

> . . . the symptoms led me to believe the problem is internal erosion of soil from under the dam. See, erosion would cause the selectively broken pipes under the dam, therefore slowing drainage and causing high uplift pressures that cause the dam to slide downstream . . .

Figure 1 illustrates the difference between this line of explanation and the original line of reasoning. The line of explanation has a clearly distinct structure and content and is not just a reformulation of the line of reasoning.

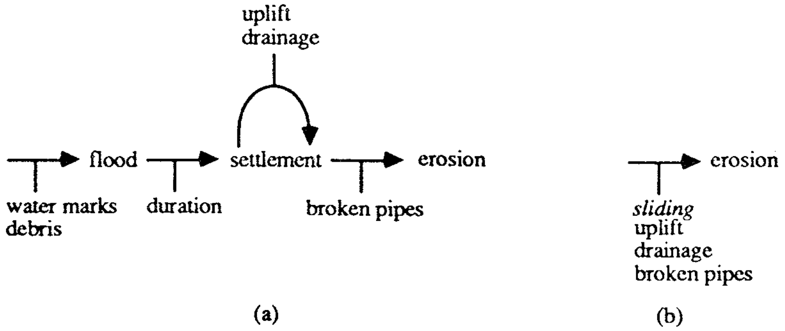

Fig. 1. The line of reasoning versus the line of explanation.

Neither augmentation with additional domain knowledge nor pruning of 'dead ends' is sufficient to generate the explanation. For example, the heuristic association between *flood* and *settlement* that directly led to the eventual conclusion *erosion* is absent from the line of explanation, even though it was an essential step on the reasoning path from symptoms to conclusion. At the same time, the line of explanation contains items not in the line of reasoning. Some involve the addition of domain knowledge describing the underlying causality of many of the reasoning steps. Others, however, introduce additional support evidence not used during the original problem solving. This includes the introduction of new symptoms that further support the final conclusion (i.e., the sliding of the dam). We call this phenomenon *data introduction* and claim that it is common in domains marked by *nonexhaustive problem solving*. In such domains, an expert will use a relatively small set of cues from the data to reach a conclusion. Once this conclusion has been made, the expert will support it with additional data items. In some explanations, the initial data cues are replaced with new more directly supporting data. In our example, the triggering data (i.e., the duration of the flood) is dropped as it is not directly related to supporting the conclusion.

This example illustrates the *decoupling* of the line of explanation from the line of reasoning. Decoupling results in a reasoning graph for explanation that is largely distinct from the reasoning graph left behind by the original problem solving. Unlike previous approaches to explanation, the distinction between these graphs can be extreme. For example, the explanation graph shown in Fig. 1 represents an entirely new movement from the symptoms to the conclusion. In fact, a completely new symptom (sliding) was added to the explanation that was not present in the original problem solving. Therefore, the line of explanation and the line of reasoning are often considerably different in both form and content. The line of explanation is no longer restricted to be just a reformulation of the line of reasoning. Our approach aims at breaking the tight bond that has previously existed in expert systems between problem solving and the explanation of that problem solving. Explanations are created as a product of a problem-solving activity largely distinct from the expert system's original problem-solving process. With this bond broken, an explanation system has the freedom to create a more naturally flowing account of the expert system's actions and conclusions.

2. Background

During the last fifteen years, numerous efforts have been undertaken in an attempt to improve expert system explanations. Three core ideas have emerged from this work [2]:

- *A trace of an expert system's execution can be used to provide an explanation of the expert system's problem solving.* MYCIN [23] was one of the first systems to explain its actions. MYCIN provided basic explanation queries including *why* and *how*. These two queries form the foundation of nearly all explanation facilities to date [30].
- *A domain model can be used to explain the rules used by an expert system.* Swartout [24] introduced a system called XPLAIN explicitly designed to attack the problem of explanation. Swartout used a *domain principle* and a *domain rationale* to record the designer's rule justification by using an automatic programmer to build the expert system. The XPLAIN system produced excellent explanations. However, the explanations lack the flexibility demonstrated in our earlier example as the line of explanation is tightly coupled to the line of reasoning. The tight level of coupling maintained by the XPLAIN system appears to be best suited for the knowledge engineer as will be discussed in Section 4.
- *Explanations can be given at different levels.* For example, Clancey [10] has built an explanation system that augments the facility provided by MYCIN. Clancey's system, NEOMYCIN, shifts the focus from the domain knowledge to the strategic problem-solving knowledge. NEOMYCIN is capable of generating *why* and *how* explanations about the strategy used to solve the problem.

Clearly, previous research has recognized the need for processing the expert system's line of reasoning before presenting it for explanation. For example, MYCIN pruned dead-end paths from the line of reasoning, Wallis and Shortliffe [27] demonstrated the advantages of pruning information from the line of reasoning based on complexity and importance, and Weiner illustrated how the justification of expert system beliefs may require reorganization of the supporting data prior to explanation [28]. However, each of these research projects views the process of explanation as the process of pruning, augmenting, or in some other way translating the expert system's line of reasoning. Decoupling, on the other hand, results in the process of explanation as being viewed as the processing of deriving a new movement and thus has the power to reinterpret data and even find additional information supporting the new line of explanation.

Recently, an exciting new development has begun to grow in expert system explanation. Explanation is no longer viewed as an add-on to the expert system's reasoning, but as a problem-solving activity in its own right. Two classes of approaches are currently being pursued. In the first, the problem solving of explanation is viewed as the rhetorical presentation of the expert system's line of reasoning (e.g., [17, 21]). In this research, the 'problem' to be solved during explanation is to construct a good rhetorical plan for presenting the line of reasoning to the user. The second views the problem solving of explanation as the construction of a justification of the expert system's conclusion (e.g., [13]). In this approach, a model of the domain is typically used to generate a causal justification supporting the conclusion given by the expert system.

Our work, while based on the same desire to treat explanation as a complex problem-solving process, takes this idea a step further. In particular, we demonstrate that the problem solving in explanation is not limited to devising a rhetorical presentation or a justification of the expert system's conclusion, but instead can involve a complete reconstruction of how the expert system reasoned to its conclusion. This leads to a fourth core idea in expert system explanation:

- *Explanation can be viewed to include a complex domain problem solving process largely distinct from the expert system's original domain problem solving process.*

The nature of an effective explanation depends heavily on the user. A knowledge engineer involved in the design and maintenance of an expert system requires an explanation facility that elaborates on precisely what the system did to accomplish a specific result. However, an explanation for an end-user is intended to increase the user's confidence in the system and/or to aid the user in understanding the consequences of the system's conclusion. A system designer clearly needs a traced-based explanation that accurately reflects the line of reasoning used in the expert system. This line of reasoning may be inappropriate, however, for an end-user. Lines of reasoning often proceed to a conclusion via obscure and indirect paths, particularly when heuristic reasoning is involved [3]. In the example presented in Section 1, an effective explanation of the conclusion *erosion*, although arrived at from a heuristic association with *flood*, may not only require a substantial reorganization of the line of reasoning, but may require the use of additional supporting information not part of the original reasoning process. Generating such an explanation is possible only within a *reconstructive explanation* paradigm in which the explanation is reconstructed by an active, problem-solving process.

It is important to realize the meaning of the term 'explanation' in our research. Clearly, the state-of-the-art in the automated explanation of expert systems involves complex problem solving. For example, several sophisticated systems have been built for user modeling, explanation planning, natural language interfaces and so on. However, each of these systems views the task of 'explanation' as the task of *taking* a trace of the expert system's reasoning (i.e., the line of reasoning) and presenting some transformation of the trace to the user. Our research, on the other hand, views 'explanation' as including the task of *determining* the reasoning trace that will be eventually transformed and presented to the user. In this sense, 'explanation' involves complex problem solving, analogous to traditional domain problem solving, that is not found in other explanation systems.

While there are obvious costs associated with the adoption of a reconstructive explanation strategy, these may not be as great as might at first be supposed. A clearer separation between problem solving and the explanation of that problem solving may reduce the need to trade off problem-solving competence for comprehensibility that often arises with conventional explanation systems [11]. An end-user often will not catch reasoning errors in a difficult to understand line of reasoning [5]. While the potential exists for inconsistencies between problem solving and the explanation of that problem solving, reconstructive explanation can aid the end-user in a better understanding of the problem and thus provide a basis for the user independently evaluating a system's actions.

3. The explanation problem

Our research is concerned with the explanation of complex problem solving. Therefore, the input to the process of explanation is a trace of some problem-solving activity. In the information processing model, this implies that the input is an information processing operation of input through process to output. Likewise, the output of explanation is a description of the input and therefore is also in the form of an information processing operation. Explanation can therefore be defined as follows:

> *Explanation*: an information processing operation that takes the operation of an information processing system as input and generates a description of that information processing operation as an output.

Figure 2 illustrates this definition. Notice that the input, and output each conforms to the 'shape' of an information processing operation. Explanation is

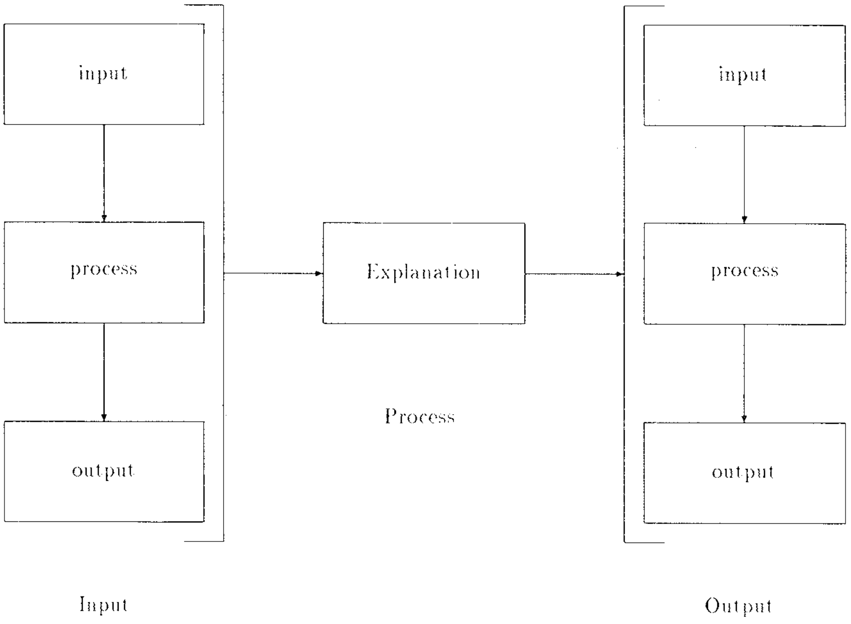

Fig. 2. Explanation as an information processing operation.

a mapping from an information processing operation to a description of that information processing operation. For example, in medicine this corresponds to observing the diagnostic process of moving from symptoms to conclusions and describing that movement to some audience.

Figure 2 shows the input and output of explanation as complete information processing operations. This is in fact not always the case. For instance, people often are able to give an explanation of an event for which they know only the output of the event to be explained. Likewise, not every explanation needs to present the complete information processing operation as a description. By investigating the nature of incomplete input and output for the problem of explanation, one can analyze the requirements of the explanation process.

3.1. Four classes of explanation

Our research is focused on the retrospective explanation of complex problem solving. That is, explanation of problem solving after that problem solving has occurred. This focus significantly constrains the completeness of the input and output of the explanation problem. A retrospective explanation is given only after the event to be explained has taken place and is known. Therefore, the output of the event to be explained must be included as input to the process of explanation. Also, an explanation of complex problem solving (i.e., of 'how') requires a movement from input through process to output. Therefore the output from the explanation process must be complete. However, the two remaining elements (the input and process) of the event to be explained may not be included as input to the process of explanation. Four distinct classes of explanation problems result from this potential incompleteness: *Output*, *Input-Output*, *Process-Output*, *Input-Process-Output*. Each class is formed by defining what elements of the event to be explained are accessible to the explanation process. For example, a common problem in medical training is to be given a physician's case, the symptoms and the diagnosis, and be asked to explain how the physician might have reached the diagnosis. In this example, the processing used by the original physician is not available, and the explainer must postulate a reasonable approach that would lead to the observed diagnosis. One might argue that such an explanation is not the same as an explanation of one's own cognitive operation. Although certainly true, Section 4 points to research which argues that both the explanation of one's own cognitive activity and the explanation of someone else's cognitive activity can require similar processing.

Our classification of explanation identifies two distinct types of problem classes. One type (*Input–Process–Output*), in which information is available on the input, process, and output of the event to be explained, simply requires a mapping from a complete information processing operation to a description of that operation. This type of problem is said to be *conveying* [31] the input via the output. The other type (i.e., *Output*, *Input-Output*, *Process-Output*), in which some information is missing from the event to be explained, requires a mapping from an incomplete information processing operation to a complete description of that operation. This type of problem is said to be *enlightening* [31] the input via the output. An example is again the case of explaining another physician's diagnosis. The key is that with enlightenment a possibly incomplete information processing operation can be mapped to a plausible description of what the complete information processing operation could have been. By plausible, it is meant that the complete information processing operation uses sound domain knowledge, but that it is not known whether it is the information processing operation that actually occurred.

The four classes of explanation represent partitions based on the access to information from the input and output. Each partition corresponds to a possible level of *coupling* between the event to be explained and the description of that event. In the least coupled level (e.g., *Output*), the input to the problem of explanation consists of only the output of the event to be explained, for example, giving a plausible explanation of another physician's diagnosis without access to the case history. In this class, the description generated by the explanation process will be only loosely bound (coupled) to the event being explained. Conversely in the most coupled level (e.g., *Input–Process–Output*) the input consists of all three elements of the event to be explained. Here, the description can be tightly bound (coupled) to the event.

3.2. Expert system explanation

Expert system explanation involves the description of a complex problem-solving process. In light of the preceding discussion, the explanation problem attacked in our research can be defined as follows:

> *Expert system explanation*: an information processing task that takes a possibly incomplete operation of an information processing expert system as input and generates a complete plausible description of that information processing operation as an output.

The audience of an explanation can significantly affect the purpose and therefore the content of an explanation. A lessened level of coupling between the event to be explained and the explanation gives the explanation process increased freedom to influence the nature of the explanation produced. This freedom can be used to alter the content of the explanation to more adequately fulfill the purpose behind giving the explanation. For an end-user audience, this purpose is to help the end-user better understand the domain in which the expert system is operating. Through enlightenment, an expert system can use the freedom of lower coupling to reconstruct a sound description of problem solving that would lead to the expert system's conclusion.

The traditional expert system explanation methodology is not powerful enough to solve problems requiring enlightenment. The traditional expert system explanation solution is to augment, prune, transform, or in some other way manipulate a complete trace of the event to be explained. This solution methodology is not capable of postulating alternative, more natural information. Certainly, support knowledge can be added, describing the knowledge found in the trace, but new knowledge or new movements through that knowledge cannot be added. For example, traditional expert system explanation methods cannot use additional symptoms to support the conclusion of the expert system.

At least two potentially significant disadvantages need to be considered when discussing decreased coupling in expert system explanation. First, in the reconstructive approach, explanation is no longer achieved simply by presenting a trace of the original problem solving. Explanation now requires additional knowledge (not just support knowledge), additional processing to construct the explanation, and additional maintenance as the explanation system is largely distinct from the expert system. Clearly there is an additional cost. However, the increase may be less than expected. We have found that the knowledge required for explanation is more accessible from the domain expert than the actual problem-solving knowledge. In fact, at least one knowledge engineering methodology is attempting to work backwards from this knowledge to discover the expert's true problem-solving knowledge [22]. However, as larger systems are built, a 'scale-up' factor will most likely come into play as more control knowledge will be needed to direct the explanation system. Related to this issue is the concern of increased maintenance cost. The issue here is the potential cost for maintaining consistency between the knowledge used by the expert system for the original problem solving and the knowledge used by the explanation system for explanation of that problem solving. The major concern is that having a largely distinct explanation system will make changing the expert system much more expensive as the explanation system will also need to be changed. We have attempted to address this question in our research by using a high-level specification for communication between the expert system and the explanation system, as will be discussed more fully later. The specification between the expert system and the explanation system is designed to help ensure that the increased maintenance is kept reasonable. The specification tells what the expert system needs to know and not how the expert system executes. As long as the specification is not violated, the operation of the expert system can be changed as often as is necessary without affecting the explanation system. The explanation system will need to be altered only when the specification is changed, a far less frequent occurrence. Further, as reconstructive explanation is intended for the end-user, the explanation system need not be built during the early, most turbulent period of expert system construction. The explanation system can be built once the basics of the expert system have been proven and when end-user explanation is considered more important.

The second major disadvantage of the reconstructive approach to explanation is the concern of user confidence. The fear is that because the explanation system does not follow the precise reasoning steps of the expert system, and does not remain "true to the expert system", the user's confidence in the expert system's conclusion might be reduced. In other words, the potential for inconsistency between the line of reasoning and the line of explanation may decrease user confidence. For a knowledge engineer or possibly a domain expert, who desires an explanation showing precisely what the system did, confidence would decrease. However, for an end-user who is not overly concerned with the internal operation, the concern is less valid. It is possible that a reconstructive explanation system might even improve an end-user's confidence. The reconstructed explanation provides an independent check on the operation of the expert system. The explanation system constructs an argument for the expert system's conclusion without being biased by what the expert system did. Thus the explanation produced serves as an independent check that the data of the case do support and lead to the conclusion that was reached. Further, as the reconstructive approach can present more direct, less obscure explanations, the end-user's understanding and therefore confidence may actually increase. Research has shown that end-users that were less skilled in the application domain were unable to use the explanation of traditional approaches [5], possibly due to their inability to follow the expert's line of reasoning. By presenting the end-user with a direct explanation that serves to increase the end-user's general knowledge of the domain, the reconstructive approach may give the end-user an increased ability to make an independent decision about the validity of the conclusion.

Overall, reconstructive explanation is not free and a decision must be made during expert system construction whether quality end-user explanations are important enough to support the extra cost. Further, the issue of inconsistency between the expert system's reasoning and the explanation of that reasoning must also be considered. Ultimately, one would desire a system that could ensure that the line of reasoning and the line of explanation stand together and fall together. In our research, we have devised a scheme of constraints that can be placed on the explanation system in order to approach the consistency guarantee. However, as the power of reconstructive explanation comes from the ability to present largely distinct movements through potentially different portions of the domain knowledge, an actual guarantee is not possible.

4. The theory

Section 3 discussed how the traditional approach to expert system explanation is too weak to solve problems requiring enlightenment. One reason that these problems may have been overlooked by previous research is that they were thought to be less useful to expert system explanation than the other problems solvable through the traditional method. In expert system explanation, a complete trace of the expert system's line of reasoning is always available. In certain situations, such a complete trace is critical to the success of the explanation facility. However, for other situations, a complete trace may serve only to limit the kind and flow of the explanation. By allowing an explanation system to fill in an incomplete trace, the system is given increased freedom. Through the power of enlightenment, the system can reorganize the explanation, allowing it to flow more directly and clearly to the final conclusion.

The utility of enlightenment depends heavily on three major features: the *goal*, *audience*, and *focus* of the explanation. The goal represents the purpose behind giving the explanation. There are three major explanation goals, namely *verification*, *duplication*, and *ratification*. Within the context of verification, the goal of the explanation system is to verify the knowledge of the expert system. Here, the explanation constitutes a description of some part of the expert system's knowledge base. A good explanation for verification is one that presents a precise, accurate, concise, and easy-to-understand description of the expert system's knowledge. In the context of duplication, the goal of the explanation system is not simply to present the knowledge of the expert system, but also to transfer that knowledge to the user. An explanation system that attempts to transfer knowledge from the expert system to the user must take care to adequately present the methods and knowledge of the expert system to the user allowing this information to be learned for future use. A duplication explanation is judged successful to the extent that it is able to transfer knowledge from the expert system to the user, allowing the user to mimic the expert system's performance. Finally, in the context of ratification, the goal of the explanation system is to increase the user's confidence in the expert system. In this light, the explanation system must concentrate on conveying an understanding of the domain to the end-user allowing for an independent evaluation of the expert system. A ratification explanation is successful to the extent that the user is comfortable with the expert system's recommendation. Note that it may be possible to ratify the conclusions of an expert system without being able to independently generate the conclusion.

The second feature that strongly influences the nature of the desired explanation is the audience. Both the content and organization of an explanation need to consider the person to whom that explanation will be given [19]. Three major audiences can be defined: *the knowledge engineer*, *the domain expert*, and *the end-user*. A knowledge engineer, when using an explanation system, is generally interested in the internal operation of the expert system. An explanation for a knowledge engineer often emphasizes 'transparency', or the ability to observe the true internal operation of the expert system. A domain expert generally uses an explanation to study what the system knows, that is the 'knowledge-level' [18] description of the system. Finally, the end-user of an expert system is most often interested in better understanding the recommendation of the expert system.

Third, the focus plays a key role in determining the required content and form of an explanation. In general, there are two main focus alternatives: the *process* and the *solution*. An explanation that focuses on the process is concerned with presenting information on the flow of the problem-solving activity. Such an explanation will in general address a question of 'how' some specific event took place. On the other hand, an explanation that focuses on the solution concerns itself with the output of the problem-solving activity. This type of explanation is commonly referred to as a justification of the conclusion. An explanation in this context will usually constitute an argument in favor of the conclusion.

These three features interact to define the nature of the explanation that is desired. For example, a knowledge engineer asking for a verification of the process will require an explanation that presents the specific details of the activity that led to the event in question (e.g., the exact rules that fired and the order in which they fired). However, a domain expert that asks for a verification of the process will require an explanation void of the internal details of the representation and instead concentrating on the domain knowledge held within that representation. This point is central to the distinction between this research and nearly all previous expert system explanation work. Previous research treats explanation as the process of presenting to the user a description of the methods used, and conclusions reached by the expert system. In our research, explanation is viewed as the process of conveying to the end-user an understanding of the domain in which the expert system is operating. This focus significantly changes the restrictions that must be placed on the explanations produced. Figure 3 illustrates the *coupling spectrum* that results from the recognition of the influence that each of the three features has on explanation. As the goal moves further from verification and closer to ratification, the level of coupling required between the line of reasoning and the line of explanation decreases. Likewise, as the audience moves further from knowledge engineer and closer to end-user, and as the focus moves further from process and closer to solution, the level of coupling required between the line of reasoning and the line of explanation decreases. Each feature suggests a particular level of coupling. For example, in giving a knowledge engineer an explanation in an attempt to verify the process used by the expert system, a tight level of coupling would be suggested. However, in giving an end-user an explanation in an attempt to ratify the process, a much lower level of coupling may be appropriate.

While the processes by which human experts produce explanations are not well understood, there is evidence that human explanations by necessity involve both enlightenment and a loose coupling between the explanation and the problem solving that actually occurred. People in fact cannot accurately explain their actual problem solving practices [6, 7]. Verbalization of internal thought processes are necessarily constrained by certain key limits of cognitive processing. Human experts must infer certain critical elements of explanations, since they cannot be aware of what actually occurred in their own thoughts. While these constraints are normally viewed as a limitation, the inferential nature of human generated explanations may actually improve the ease with which they can be understood by others. With expert system explanation, a complete trace of problem solving is of course available. Nevertheless, explanations generated by inference from key aspects of the trace rather than from detailed histories may be beneficial. As with the case of human experts, such explanations may be better understood by the intended audience.

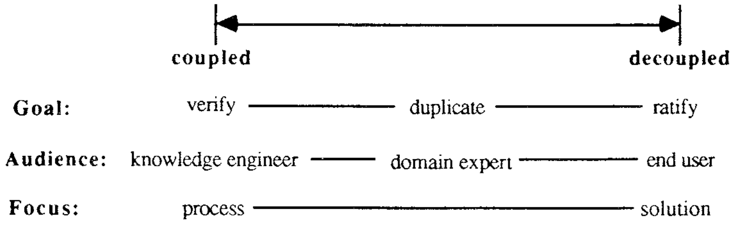

Fig. 3. The coupling spectrum.

5. The model and its implementation

This section defines a model of expert system explanation capable of using the power of enlightenment to tailor an explanation to an end-user audience. There are four central ideas that support this model. First, a clear interface must exist between the expert system and the explanation system. In the reconstructive model of explanation presented here, this interface is given by the knowledge specification of the expertise required to solve problems within the application domain. Second, variations in coupling between the line of reasoning and the line of explanation must be possible. These variations allow the goal, audience and focus to influence the desired nature of the explanation as discussed in Section 4. A taxonomy of constraints is used to specify the required correspondence between the details of problem solving and the generated line of explanation. Third, there must be a body of knowledge that can be used to account for the information uncovered by the expert system. This body of knowledge should represent natural, easy-to-understand, sound relations between the items used by the expert system. Note that this information is not necessarily the information that should be used during the original problem solving. It is simply a set of knowledge that can be used to convey an understanding to the user. This is analogous to saying that the best way to solve a problem may not be the best way to explain the solution to the problem. Through enlightenment, the information in this explanatory knowledge base is used to describe and account for the actions and conclusions of the expert system. The resulting description follows the cleaner, more direct flow of the explanatory information while still presenting a sound movement from data to conclusion. Fourth, the explanation presented to the user should follow the flow of an information processing operation as discussed in Section 3. The explanation should be structured as a story that moves in a natural manner from the initial problem state to the final conclusion.

REX (reconstructive explainer) is a fully-implemented prototype explanation system built to show the feasibility of using the reconstructive approach to expert system explanation. REX is designed to provide explanations of how an expert system moved from the data of a case to the final conclusion. In other words, it provides an explanation of the movement in the competitor set of final hypotheses from the initial problem state to the final conclusion reached by the expert system. The REX program has been used with two expert systems. One deals with the design of experiments to study the relationship between factors of some industrial process (see [29]). The second, illustrated below, comes from the domain of risk analysis of an existing concrete gravity dam (based on [9]). Each is an example of a classification system. In the risk analysis domain, the dam is classified according to the most likely cause of a high reservoir load. In the experimental design domain, the expert system operates by first classifying the problem as requiring a particular type of experiment and then refines that general classification to give a specific instance of the experiment appropriate for the particular problem at hand. Therefore, REX as it currently stands is restricted to explaining classification expert systems. This restriction appears to be a result of our test domains and not a result of any inherent limitations of the REX design.

5.1. A reconstructive explanation model

Figure 4 presents an overview of the model of reconstructive expert system explanation used in this research. The *expert system* produces a set of *reasoning cues* that represent key data and inferences used during the original problem solving that moved the expert system to the conclusion. These reasoning cues are passed through a *screener* that, based on certain *problem constraints*, determines the information from the reasoning trace of the expert system that is available to the explanation system. These constraints force the explanation system to solve one of the four classes of explanation problems defined in Section 3. The cues that survive the screening process are mapped into a

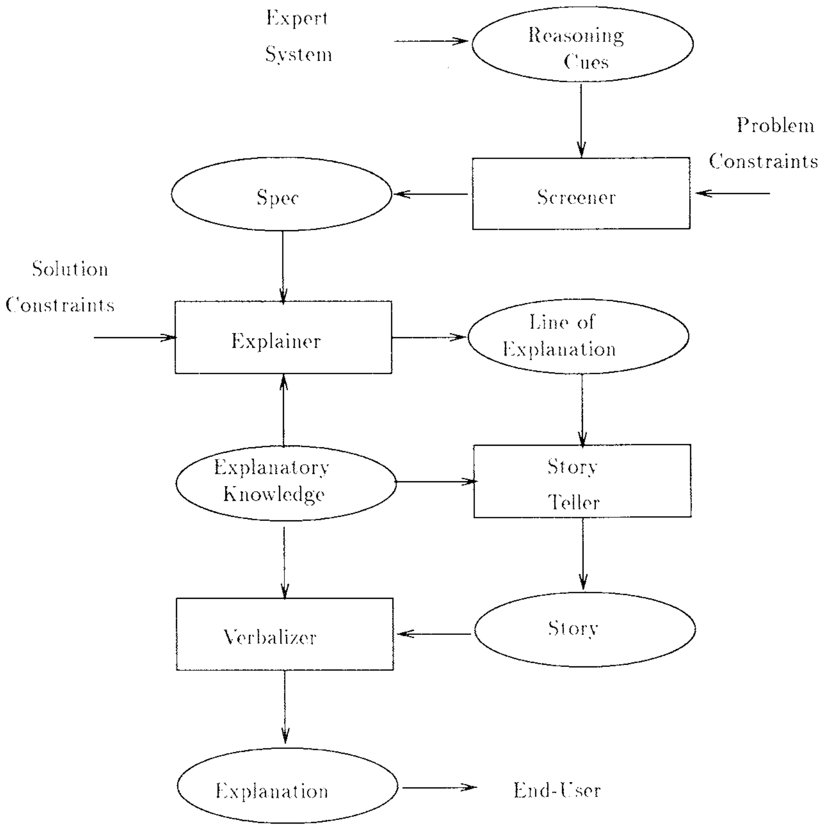

Fig. 4. An overview of the explanation model.

knowledge specification of the domain in which the expert system is operating. The specification provides an additional filter of the details of the expert system's reasoning. The specification, tailored to the particular case by the reasoning cues, is then passed to the *explainer* along with certain *solution constraints* that influence the level to which the explainer must follow the expert system's reasoning cues in the specification. The explainer attempts to enlighten methods and relations from the domain *explanatory knowledge* that account for these reasoning cues. Within the solution constraints, the explainer finds an explanation that leads from the initial data to the final conclusion. The *line of explanation* is passed to a *story teller* that formats its structure to follow that of an information processing operation. The resulting *story* is then examined by a *verbalizer* that translates the story into the final *explanation* to be given to the *end-user*.

5.2. The REX *system*

This section discusses how each element of the above model is implemented in REX. Overall, REX can be viewed as searching a restricted subset of the explanatory knowledge base in order to find a line of explanation supporting the expert system's reasoning. REX takes three parameters: the reasoning cues of the expert system, the problem constraints, and the solution constraints. These three parameters dictate the level and nature of the coupling between the line of reasoning and the line of explanation. This section will illustrate the REX system by walking through a sample execution and describing the processes and structures used in reconstructing the final line of explanation.

Before describing the implementation of the REX system, it is necessary to explicitly address how REX is designed to fit into an overall explanation interface for an expert system. The REX system, as it is currently implemented, is intended to be one component of an overall reconstructive explanation interface to an expert system. In particular, REX is a tool that can reconstruct explanations of various kinds depending on the nature desired by the interface system as a whole. The decisions as to what kind of explanation is required for the particular end-user of a reconstructive explainer are not made within the REX system. The ultimate goal is to have other components of a reconstructive user interface, such as a user modeling system and a dialogue manager, that could be used to automatically determine the kind of explanation most appropriate. However, as it now stands, decisions on such important matters as what level of coupling should exist between the line of reasoning and the line of explanation, and what kind of explanation (i.e., shortest, least complex, etc.) should be generated, are left to be answered by the person who is constructing the explanation system for the particular expert system application. Our discussions give indications as to what answers may be appropriate, however much research is still needed (see Section 7).

5.2.1. Input to the REX *system*

There are three input parameters to the REX system: the expert system's reasoning cues, the problem constraints and the solution constraints.

The reasoning cues

The expert system is responsible for the construction of the original line of reasoning. Within our model, explanation is constrained to be retrospective and thus the expert system is assumed to have completed a line of reasoning from the initial problem state to some final conclusion. For our demonstration example, we will use an expert system of risk analysis for concrete dams intended to aid a field inspector in determining the potential risk for the release of water. The particular problem is to identify the cause of an excessive load on the dam. In other words, the dam measures an excessive amount of pressure from the water in the reservoir. The field inspector is to determine what has caused the increased pressure. The expert system is run and a trace of its line of reasoning is maintained, representing the inference steps that led the expert system from the initial data of the case to the final conclusion. For our demonstration example, these inference steps correspond to the line of reasoning shown in Fig. 1(a). The reasoning cues are extracted from the expert system's line of reasoning to allow the explanation system to follow, to the extent desired, the authentic operation of the expert system. The explanation system uses the cues from the line of reasoning rather than the complete line of reasoning so as to separate 'what' the expert system did from 'how' it was done. A reasoning cue can be one of two types: direct or indirect. A direct cue represents information found directly in the case being solved. For example, symptoms in our risk analysis domain. An indirect cue represents some intermediate conclusion or hypothesis that was reached based on one or more other cues. The line of reasoning in our example leads to the following list of cues: (*water-marks debris preconditions drainage uplift broken-pipes structural-conditions*). The relationships between these items, including inference sequence, data dependence, and temporal ordering are not preserved.

The problem constraints

Following the coupling spectrum in Fig. 3, it is necessary to allow various degrees of coupling between the line of reasoning and the line of explanation. A natural way of doing this is to allow constraints on how closely the line of explanation must follow the line of reasoning. In the REX system, we have used two sets of constraints. The first set of constraints, the problem constraints, act to screen the reasoning cues obtained from the expert system. The second set of constraints, the solution constraints, define how closely the REX system must adhere to the reasoning cues which survive this screening process. As is discussed later, there is some redundancy in the use of two sets of constraints, however, they provide a natural means of specifying which elements of the expert system's reasoning are important and how closely the line of explanation must adhere to those elements.

The problem constraints determine the kind of reasoning cues passed from the expert system to the explanation system. The reasoning cues meeting the problem constraints will be matched against the solution constraints introduced later in order to determine the level of coupling between the line of explanation and the line of reasoning. In REX, four possible problem constraints (*No-Restrict*, *Direct-Only*, *Indirect-Only*, and *Direct-Indirect*) are used corresponding to the four classes of explanation problems described in Section 3. As noted earlier, complete information is always available in expert system explanation. However, the use of reconstructive techniques may be partly responsible for higher quality explanations. As such, we seek to have the REX system use enlightenment to explain the actions of the expert system. To control the level of enlightenment required, REX uses the problem constraints to artificially construct one of the four explanation problems introduced in Section 3. This gives the REX system the ability to vary the amount and kind of coupling that will be possible between the line of reasoning and the line of explanation. For example, the problem constraint *Indirect-Only* sets the scope of coupling to include only indirect cues. In other words, the explanation system can only be constrained to follow indirect cues from the expert system's line of reasoning. In REX, each reasoning cue is marked as either direct or indirect. The problem constraints prune cues from the set of reasoning cues based on this marking. In the example presented in this section, the problem constraint *Direct-Indirect* will be used so that complete coupling to the cues of the line of reasoning is possible. Thus both the traditional approach and the reconstructive approach to expert system explanation are capable of producing an explanation for this case.

The solution constraints

The solution constraints act to control the amount of freedom given to the explanation system. They are designed to allow an incremental loosening of the coupling between the line of reasoning and the line of explanation. In this way, the solution constraints allow discrete points to be chosen along the coupling spectrum introduced in Section 4 (the choice of what solution constraints to use is discussed later). These constraints allow approximations to the guaranteed consistency between the line of reasoning and the line of explanation. Although a complete guarantee is not possible in the reconstructive paradigm, it is possible to guarantee consistency on certain aspects. For example, one might wish to constrain the explanation system to only consider lines of explanation that use exactly the same direct cues as were used by the expert system. In such a case, the data presented in both the line of reasoning and the line of explanation would be guaranteed consistent. Although not a guarantee of total equivalence, it is nonetheless a stronger assurance than if no constraints were used at all.

In Rex, there are five possible solution constraints corresponding to five points along the coupling spectrum starting at the least coupled end and progressively moving towards the most coupled end: *No-RC*, *RC*, *Only-RC*, *All-RC*, and *All-Only-RC*. Each constraint dictates the degree to which the Rex system must follow the reasoning cues (RC) highlighted in the knowledge specification. The constraint *No-RC* allows Rex to ignore the reasoning cues from the expert system and thereby reconstruct for itself any path it wishes to use for the line of explanation. *RC* requires that Rex only consider lines of explanation in which the hypotheses visited are directly supportable with the reasoning cues used by the expert system. *Only-RC* requires that the candidate lines of explanation only use the reasoning cues used by the expert system. In other words, no additional cues may be added by the Rex system. *All-RC* constrains Rex to consider only lines of explanation that contain all of the reasoning cues used by the expert system. Finally, *All-Only-RC* constrains Rex to following all of the reasoning cues and does not allow the introduction of any new reasoning cues. This last solution constraint represents the tightest coupling supported by the Rex system.

In implementing the above constraints, a distinction is made between local and global constraints. The solution constraints *No-RC*, *RC*, and *Only-RC* are local constraints as they can be enforced on the entire line of explanation by enforcing them on each transition between hypotheses in that line of explanation. However, the solution constraints *All-RC*, and *All-Only-RC* are global constraints in that enforcing them on each transition will not necessarily enforce them on the entire line of explanation. The problem is that these two constraints force the Rex system to account for *all* the reasoning cues. As such, each complete line of explanation must be checked for conformance with these constraints. In fact, as Rex follows the general strategy of not presenting dead-end reasoning in the final line of explanation, it may not be possible for any single line of explanation to satisfy these constraints. In this case, the line of explanation most closely satisfying the solution constraint is presented as the final line of explanation.

Clearly, there is an interaction between the problem constraints introduced earlier and the level of coupling enforced by the solution constraints. The problem constraints dictate the classes of reasoning cues to which the solution constraints will be applied. For example, the problem constraint *Direct-Only* limits the influence of the solution constraints to the direct cues only. Therefore, the tightest level of coupling (i.e., the *All-Only-RC* solution constraint) within this problem constraint will simply enforce that Rex only consider lines of explanation that account for all of the direct reasoning cues used by the expert system and not allow the introduction of any additional direct reasoning cues. Obviously the problem constraints implemented in Rex are not the only possible choices. A reconstructive explainer could be implemented with no notion of problem constraints. In this context, the solution constraints would apply equally to every reasoning cue taken from the expert system. For example, the solution constraint *All-Only-RC* would force the explanation system to account for every cue (direct and indirect) used by the expert system. In such a situation, it would not be possible to instruct the explanation system to construct a line of explanation that uses the same data (direct cues) as the expert system but that presents a different strategy for reaching the conclusion. Such an ability to present alternative explanations is often critical in explaining a solution to different users. Another choice might be to individually mark each reasoning cue as to whether it should be considered during explanation. Although this would give complete control over the content of the line of explanation, it would be a tedious and time consuming method of specifying the overall constraints on the system. In REX, we have chosen a compromise set of problem constraints that allow a distinction between direct cues (data) and indirect cues (hypotheses and conclusions) without needing the explicit enumeration of each reasoning cue.

The question remains concerning how one chooses what problem and solution constraints to use for a particular execution of the REX system. As the system stands now, this decision is left to the designer of the explanation system. Our hypotheses concerning the relation between the spectrum of coupling and the type of user requesting the explanation give a rough estimate as to what choices might be appropriate. However, much research is still needed to more systematically assign constraints to user classes or even to particular users. For our demonstration example, the solution constraint *No-RC* will be used to illustrate the most extreme level of decoupling.

5.2.2. The knowledge base

There are two key elements to the knowledge base: a specification of the expert system's knowledge base and a separate explanatory knowledge base used only by the REX system.

The knowledge specification

The knowledge specification defines the interface between the knowledge used in the expert system and that used in the explanation system. This specification is used to represent the 'common ground' between the knowledge base of the expert system and the knowledge base of the explanation system. The basic idea is to use a high-level specification of the domain so that the details of the expert system's reasoning can be ignored while maintaining the essence of what the expert system has done. In the REX implementation, the specification is represented as a graph of potential solutions or hypotheses along with all information that might lead to transitions between these hypotheses [12].[1] Figure 5 shows the graphical representation for a portion of the specification in our example domain. The interpretation of this graph is that any transition between two hypotheses (shown in ovals) will require the satisfaction of a subset of the goals found in the cloud using cues found in the box, relying on the relations and abilities found in the triangle (in Rex, the information found in these triangles could be used for user modeling, but this has not been implemented in the current version). Notice that this representation is neither procedural or deterministic, thus representing the 'what' of problem solving without representing the 'how'. For example, many paths exist from the initial empty hypothesis (i.e., "=" in Fig. 5) to the hypothesis *settlement*. Further, each path to *settlement* has several alternatives for the data cues and strategies used in traversing that path. Each transition holds all knowledge required to move between the two hypotheses, but does not tell what subsets of this knowledge are to be used in each instance of a transition between the two hypotheses. That is, the specification represents the general knowledge of the domain showing what information can be used to move between different hypotheses. The solution to a particular problem represents a choice of exactly what knowledge will be used in moving through the specification. Alternative problem-solving strategies will result in alternative subsets of the knowledge in the specification being used. The distinction between the 'how' and the 'what' of problem solving allows Rex to follow what the expert system did without being constrained to follow precisely how it was done.

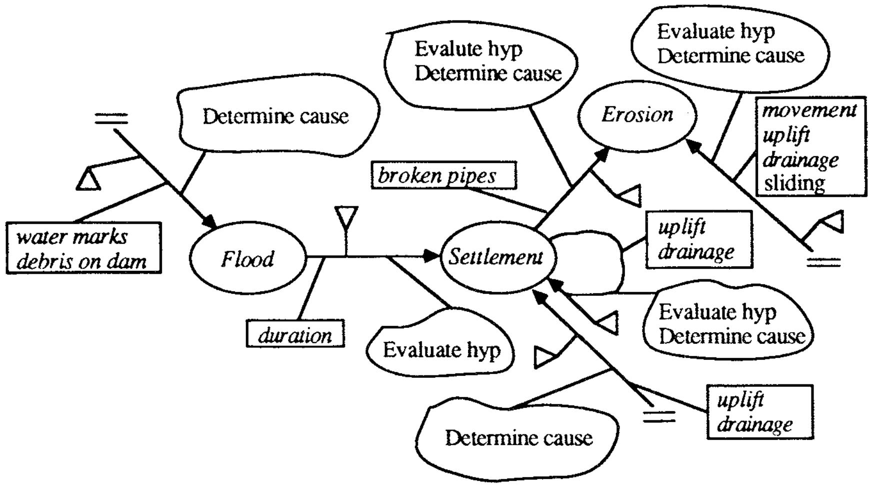

Fig. 5. A partial knowledge specification for the risk analysis domain.

[1] The process of constructing such a specification is not yet well-understood. Johnson et al. give one possible technique [12], but much research is still needed.

In REX, each cue surviving the screening process is 'highlighted' (activated) in the knowledge specification. Each reasoning cue not corresponding to an element of the specification is dropped from consideration. For example, details of the expert system's problem solving such as (*preconditions structural-conditions*) are not considered. The result is the general knowledge specification in which the elements used by the expert system's line of reasoning have been highlighted. This corresponds to the problem-solving process of the expert system being overlaid onto the general knowledge of the domain. As a result, both the general knowledge of the domain and elements of the specific knowledge used by the expert system are available to the REX system. The resulting specification for our demonstration example is given in Fig. 5 with the highlighted reasoning cues shown in italics.

The explanatory knowledge base

The explanatory knowledge base is central to the explanation process. Research on human problem solving has recognized that there is a distinct difference between the type of knowledge used for the original problem solving, and the type of knowledge used to describe that problem solving [26]. For example, medical students are often presented information in the form of *symptoms given disease*. In practice, most medical problems are solved as *disease given symptoms*. The reason for this difference is that the knowledge presented in textbook explanations critically depends on the solution being already known. An author, when writing a textbook, knows where the reasoning is leading and can therefore present an elegant method of moving from the symptoms to the disease. However, this prior knowledge is seldom available during the original problem solving, requiring reasoning that is often more obscure and indirect. The model of reconstructive explanation presented in this report strives to use the explanatory knowledge of a domain to account for the cues found during this more obscure reasoning process. As a result, sound methods and relations can be presented, giving a more natural movement from data to conclusion.

In our research, we have found the explanatory knowledge base to have two main distinctions from the knowledge base of the expert system. First, the explanatory knowledge base has a more general structure than that of the expert system's knowledge base. In particular, the explanatory knowledge base lacks the kind of 'train of thought' restrictions of the expert system. The knowledge base of the expert system is slanted towards problems solving and as such has compiled interpretations. The explanatory knowledge base holds a deeper representation showing more general relations and alternative interpretations not present in the expert system's knowledge base. Second, the explanatory knowledge base contains idealized methods (strategies) for solving problems that come close to resembling 'textbook' style presentations of the domain. In contrast, the expert system's knowledge base is composed of the typical convoluted set of intermixed control and domain knowledge. In addition, the explanatory knowledge base, as it is designed for explanation, has several alternative strategies and data interpretations that are not necessary for the expert system's problem solving. Overall, the two knowledge bases appear to be related by a selective compiling of the explanatory knowledge base to give the expert system's knowledge base. It is worth noting that in our research we have found that even when the two knowledge bases are very similar in both structure and content, the ability to create new movements through the knowledge for explanation can still be a very powerful tool. For example, a particular end-user may prefer certain elements of the knowledge base to others and would therefore prefer an explanation that used those elements over an equivalent explanation that did not (see Section 6 for an example). The only restriction in Rex is that the two knowledge bases agree on the knowledge represented by the specification described earlier. This means that the expert system and the explanatory system must at least share the goals, cues, and hypotheses listed in the specification. This ensures that the explanation system has a proper context for reasoning about the actions of the expert system, as they both share a common language of domain knowledge.

The explanatory knowledge base is represented in Rex as a collection of relationships between data cues, hypotheses, and goals. Each data cue, representing a domain noun such as *uplift pressures*, and each hypothesis, representing a potential conclusion such as *flood* or *settlement*, is represented in Rex as illustrated in Fig. 6.[2] Each cue is represented as a frame with five major slots. Each cue has a *value* and is classified by *type* as either direct or indirect to allow the problem and solution constraints to appropriately define the level of coupling. Also, each cue has a *name*, a *nickname*, and a *valuename*. These fields are used to store the text describing each particular cue. The nickname is used whenever possible to reduce the length of the resulting explanation. The valuename presents a cue and its value. Each cue of the risk analysis example is propositional, therefore the valuename is simply the cue's name itself. The hypothesis frame has no type but uses the other slots of the cue representation.

Figure 6 also shows how each goal used in moving between two hypotheses is represented in Rex. Again, each goal is represented as a frame, this time with two slots. Just as with the cue and hypothesis representations, a *name* and *nickname* are given to each goal.

A relationship in the explanatory knowledge base is represented as a script relating cues, hypotheses, and goals. Figure 6 shows an example cue and goal script. Each cue script is equipped with a list of the cues it uses, the hypothesis it supports, the bottom (leaf) cues it assumes, as well as the goal it can achieve. For example, the cue script shown in Fig. 6 achieves the goal *det-cause* as it

[2] This representation was chosen to ease the task of text generation in the prototype. Richer structures will likely be required for more complete explanations systems.

```
CUE uplift
   Value        : true
   Type         : direct
   Name         : the high uplift pressures acting on the dam
   Nickname     : uplift pressures
   Valuename    : *cue*

HYPOTHESIS erosion
   Value        : true
   Name         : the erosion of soil from under the dam
   Nickname     : erosion
   Valuename    : *hypothesis*

GOAL det-cause
   Name         : determine causal relationships
   Nickname     : determine causes

CUE SCRIPT erosion-to-sliding
   Uses         : (<drainage> <uplift> <sliding>)
   Supports     : <erosion>
   Achieves     : det-cause
   Bottoms      : (<drainage> <uplift> <sliding>)
   Vconstraint  : (and <drainage> <uplift> <sliding>)
   Text         : (<erosion> would cause <broken-pipes> resulting in <drainage> thereby creating
                  <sliding> and eventually <uplift>)

GOAL SCRIPT causal
   Holds        : (<det-cause>)
   Text         : (simply <det-cause>)
```

Fig. 6. Sample explanatory knowledge representations.

explicitly identifies causal information for the risk analysis domain. The 'vconstraint' defines the condition that enables a cue script to be used. For cue script *erosion-to-sliding*, each cue must be true in order for a cue script to be used. Each goal script represents a method from the explanatory knowledge base. These methods form the procedural knowledge of how to move between hypotheses. Each script has a text slot for presentation.

Using the representation described above, a transition from one hypothesis to another is possible when methods using only goals listed in the 'goal cloud' of the transition edge are found such that each goal in each method is achieved by relationships using only cues found in the 'conditions box' of the edge. In REX, the structure built by combining the methods and the relationships is called an *explanation structure* as it serves as an explanation of the movement between the hypotheses. Figure 7 illustrates the explanation structure con-

```
EXPLANATION STRUCTURE
   Name       : ES-101
   From-Hyp   : nil
   To-Hyp     : HYPOTHESIS erosion ...
   Methods    : ( (GOAL SCRIPT causal ...) )
   Relations  : ( (CUE SCRIPT erosion-to-sliding ...) )
   Showns     : ( (CUE uplift ...) (CUE drainage ...) )
   Bottoms    : ( (CUE uplift ...) (CUE drainage ...) )
```

Fig. 7. A sample explanation structure for the risk analysis domain.

structed for our demonstration example. The following paragraphs will describe how this explanation structure is created.

5.2.3. The processes

Four main processes are used to reconstruct the line of explanation: the screener, the explainer, the story teller, and the verbalizer.

The screener

The screener is responsible for pruning each element from the reasoning cues that does not meet the problem constraints. In our example, the screener returns the untouched list (*water-marks debris preconditions drainage uplift broken-pipes structural-conditions*) as the problem constraint *Direct-Indirect* is being used.

The explainer

The explainer is responsible for searching the explanatory knowledge base to find relations and methods that account for the information highlighted in the specification. This search is carried out within the restrictions imposed by the problem and solution constraints. The result is the line of explanation that will eventually be presented to the end-user. This line of explanation represents a movement from the initial problem-solving state to the final conclusion reached by the expert system.

In Rex, the line of explanation corresponds to a path through the knowledge specification from the conclusion of the expert system to the empty hypothesis. Each transition in this path must be supported by the existence of an explanation structure that uses only cues satisfying the solution constraints. For example, when Rex is not allowed to introduce any additional information (i.e., *Only-RC*), valid explanation structures can only use those cues contained within the line of reasoning. However, this restriction only applies to the reasoning cues that satisfy the problem constraints. Therefore, in the context of the problem constraint *Indirect-Only*, the Rex system has freedom to include any additional direct cues that it can find in an attempt to support the indirect cues found in the line of reasoning. Thus, in this example, the Rex system must use only the indirect cues found in the line of explanation, but can use additional direct cues to support the indirect cues above and beyond those used by the expert system.

Following these restrictions, the explainer attempts to build valid explanation structures for each hypothesis transition under consideration. The elements of the explanation structure are filled in by inspecting the explanatory knowledge base in order to find any goal scripts that support the *To-Hyp* hypothesis of the current transition. However, this inspection must be consistent with the knowledge specification in that the goal script must only use goals listed in the 'goal cloud' of the transition edge. Likewise, the explanatory knowledge base is also inspected for cue scripts that achieve the desired goals using only cues listed in the 'conditions box' of the transition edge. Cue scripts represent fossilized inference relations in the explanation knowledge base. As such, they establish certain cues as conclusions and use other cues as data. When the data cues used by a cue script are not direct cues, other cue scripts must be found that establish these non-terminal leaf cues. This establishes relationships between the cue scripts of an explanation structure that must be considered when presenting the explanation structure to the end user (see the verbalizer discussion for details). However, as the only cues that can be used are those listed in the 'conditions box', some of the non-terminal leaf nodes may not be supportable in the current transition. These nodes, along with the terminal leaf nodes (i.e., the leaf nodes which are direct cues) become the *Bottoms* of the explanation structure. The non-terminal *Bottoms* will need to be supported by explanation structures built for transitions to hypotheses earlier in the line of explanation.

The task faced by the explainer is to find a path through the knowledge specification using only transitions for which a valid explanation structure can be constructed. REX uses the A* algorithm[3] to search through a space of knowledge specification transitions for which a valid explanation structure has been found. The search is carried out backwards from the final conclusion of the expert system towards the empty hypothesis. Each state in this search corresponds to an emerging line of explanation that uses certain cues and a hypothesis (the *Bottoms* and the *From-Hyp* hypothesis) as data, establishes other cues and a hypothesis (the *Showns* and the *To-Hyp* hypothesis) as conclusions and traverses certain edges in the knowledge specification. Operators in the A* search correspond to expanding the *From-Hyp* hypothesis by finding each transition edge in the specification with a valid explanation structure that moves to this hypothesis. As the precise explanation structure chosen will determine the cues included in the explanation, a separate transition in the A* search is constructed for each valid explanation structure on each incoming edge of the bottom hypothesis. A complete line of explanation is found when the *From-Hyp* hypothesis is the empty hypothesis and all *Bottoms* cues are direct cues.

The cost of a state in the A* search is given by the number of cues *Shown* in the corresponding line of explanation plus the total cost of the transition edges included. The heuristic function used in REX represents the number of cues remaining to be shown (non-terminal *Bottoms*) less a reward for each of these cues that was used by the expert system. In this way, the search is directed not only towards the most direct line of explanation, but also towards the most direct line of explanation that stands the best chance of being successfully

[3] The use of the A* algorithm is simply to show feasibility. A more complete system would possibly need to use a more advanced search strategy.

constructed. In the risk analysis example, the line of explanation found uses the cue script of Fig. 6 to achieve the strategy of determining causal relationships, thus allowing a movement from the empty hypothesis to erosion.

By altering the cost and heuristic functions of the A* procedure, the explainer can be given different viewpoints that determine the best line of explanation to pursue. For example, the viewpoint can be to find the most direct line of explanation leading from the initial data to the final conclusion (i.e., fewest hypothesis transitions). This will cause the explainer to find the shortest valid path to the conclusion, thereby finding the most direct line of explanation. By changing the viewpoint of the explainer, alternative explanations will be found. As discussed earlier, the decision as to what viewpoint is most appropriate, although ultimately made by an intelligent user interface system is currently left to the designer of the particular REX application.

The line of explanation represents the movement from data to conclusion that will be presented to the end-user. Dead-ends are not included in this movement [15, 16]. The line of explanation found by the explainer for our demonstration example is simply the transition from the empty hypothesis to *erosion* supported by the explanation structure shown in Fig. 7.

The story teller

The story teller is responsible for taking the line of explanation found by the explainer and organizing it into a coherent flow from data to conclusion. The line of explanation is presented as a 'story' of how the expert system used cues to move through a series of hypotheses from the initial problem state to the final conclusion produced by the expert system. By presenting the movement as a story, the explanation system is able to structure the explanation to allow an easy and natural transfer of information from the line of explanation to the end-user audience.

Figure 8 shows the grammar used in the REX system for producing such a story. The grammar is based on research in psychology that identifies the

```
LOE            ::= {problem description} {problem statement} Story Resolution
Story          ::= Setting Theme Plot Resolution |
                   Setting Theme Plot Resolution Story
Setting        ::= {hypothesis}
Theme          ::= Event Thematic-goal
Event          ::= {cue} | {cue} Event
Thematic-goal  ::= {make initial hypothesis} | {refute hypothesis} | {generalize hypothesis}
                   {support hypothesis} | {refine hypothesis}
Plot           ::= Strategy Relations Outcome
Strategy       ::= {goal} | {goal} Strategy
Relations      ::= {explanatory relation} | {explanatory relation} Relations
Outcome        ::= {hypothesis}
Resolution     ::= {hypothesis}
```

Fig. 8. The grammar used by REX.

memory structure built during human story understanding [14, 25]. The basic idea is to extract from the explainer's line of explanation information concerning the structure of (story behind) each hypothesis transition, including the hypothesis from which the transition was made, the data cues and transitional goal which motivated that transition, the strategy used in achieving that transition, and the eventual hypothesis to which the transition was made. To complete the story, a problem description, and a problem statement are added to the front of the transition stories to serve as an introduction to the line of explanation. Likewise, the final conclusion of the expert system (the resolution of the overall line of explanation) is explicitly added to the end. The problem description and statement are found in the explanatory knowledge base as canned presentations of the general domain and problem being faced by the expert system. The resolution is given by the expert system's conclusion.

Following the grammar of Fig. 8, each transition between two hypotheses is formatted as a story, complete with a setting, theme, plot, and resolution. The setting of each story is the hypothesis from which the transition is being made. The resolution is thus the hypothesis to which the transition is made. This represents the motion of thinking about one hypothesis and considering certain data leading to another hypothesis. The goals used during this transition form a strategy that leads from the setting to the resolution. Each strategy relies on two other pieces of information: an event and the relations between the elements of the event. The event represents the data cues (either direct or indirect) that motivated the transition from the setting hypothesis to the resolution hypothesis. As such, the bottom cues of the explanation structure linking the setting to the resolution make up the motivating events. In other words, these bottom cues are what led to the movement from the setting hypothesis to the resolution hypothesis.

Figure 9 illustrates the story tree constructed by overlaying the grammar of

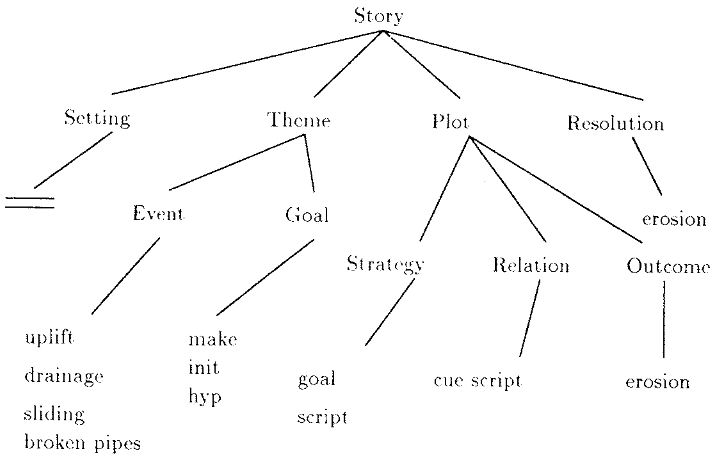

Fig. 9. The story tree.

Table 1
The goal lookup table.

${}_{\text{setting}}REL_{\text{resolution}}$	Thematic-goal
$s = \emptyset \wedge r \neq \emptyset$	Make an initial hypothesis
$s \subset r$	Generalize the setting hypothesis
$s \supset r$	Refine the setting hypothesis
$s = r$	Support the setting hypothesis
$\emptyset \neq s \neq r \neq \emptyset$	Refute the setting hypothesis

Fig. 8 onto the line of explanation for our demonstration example. This story tree is constructed as follows. The setting is given by the hypothesis from which the transition is being made. In this particular case, the setting is the initial problem state (the empty hypothesis). The theme of the story tree is given by the events that led to the thematic-goal of moving between the two hypotheses. The events, as described earlier, are the bottom cues used in making the transition. Since the story represents a transition between two hypotheses, the thematic-goal is the general relationship between these two hypotheses as defined in Table 1. In this example, the setting hypothesis is empty and the resolution hypothesis is non-empty. Therefore, the thematic-goal is to make an initial hypothesis.

The plot of the story is given by showing how the thematic-goal was achieved. The plot relates each cue, goal, and relation that was used to lead to the story's outcome. The goals of the explanation structure form the strategy followed in making the initial hypothesis. In the risk analysis example, the strategy is to determine causal relationships between the observed cues. The relations that carry out this strategy are given in the explanation structure's cue scripts. The resolution of the story is the supported hypothesis. Each transition in the line of explanation is translated into a story as described above. In the risk analysis example, there is only one such transition. The complete line of explanation composed of the problem description, the problem statement, the transition story, and the final resolution is illustrated in Fig. 10.

The verbalizer

The verbalizer is responsible for the presentation of the story tree to the end-user. In REX, this presentation involves the following activities: *describe the problem*, *describe the goal*, *describe the movement*, *and describe the conclusion*. Three of these activities are straightforward. The description of the problem and goal are given by looking up two stored descriptions in the explanatory knowledge base. The description of the conclusion simply introduces the conclusion reached by the expert system. The template for a complete story tree is shown below:

> {Problem Description}. I attempted to find {Goal Description}. {Movement Description}. Feeling confident in this solution, I concluded that {conclusion} was {Goal Description}.

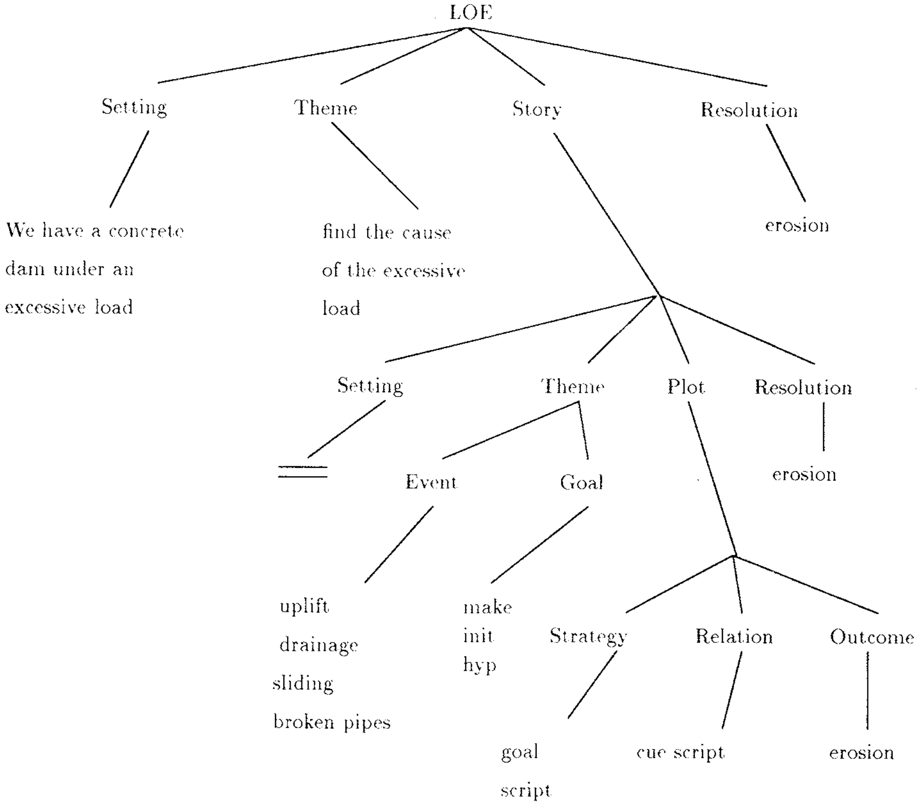

Fig. 10. The complete story tree.

Partial instantiation of that template in the risk analysis example leads to:

> We have a concrete dam under an excessive load. I attempted to find the cause of the excessive load. {Movement Description}. Feeling confident in this solution, I concluded that the erosion of soil from under the dam was the cause of the excessive load.

The description of the movement leading from the data to the conclusion requires the verbalizer to translate each story tree illustrated earlier into a text presentation. Each story tree is translated to English text following four main activities: *describe the setting*, *describe the theme*, *describe the plot*, and *describe the resolution*. These activities result in the following template used to expand {Movement Description}:

> {Setting Description} and {Theme Description}. {Plot Description}. {Resolution Description}.

The setting description is determined by the type of value found in the setting of the story. If the setting is empty (initial problem state), the setting is described as "Not knowing the solution". Otherwise, the setting is described as

"Thinking the solution might be {hypothesis}". The theme description is the presentation of each bottom cue that led to this story along with the goal that this story is attempting to achieve. This description follows the form "based on {event}, I was able to {goal}". The resolution description is the presentation of the outcome of this particular story. This description follows the template "This led me to hypothesize that {outcome} was {Goal Description}". Thus, the {Movement Description} becomes:

> Not knowing the solution and based on the broken pipes in the foundation of the dam, and the downstream sliding of the dam, and the high uplift pressures acting on the dam, and the slow drainage of water from the upstream side of the dam to the downstream side I was able to make an initial hypothesis. {Plot Description}. This led me to hypothesize that internal erosion was the cause of the excessive load.

To translate the {Plot Description} portion of the text, the following activities are followed: *describe the strategy*, *describe the relations*. The outcome description is not repeated as it follows immediately after the presentation of the plot structure. To describe the strategy, an explanatory method that relates the goals of this particular plot is presented. This presentation is structured as "To achieve this, I used the strategy of striving to {goals}". Each explanatory relation achieving each goal of the strategy is presented as "In attempting to {goal}, I found that {relations}." In presenting the relations, attention must be paid to the order in which the cue scripts are considered. As described earlier, certain cue scripts can establish indirect cues that are assumed by other cue scripts. Clearly, all cue scripts that establish indirect cues for another cue script must be presented before the cue script that assumes their existence. Completing the expansion of {Plot Description}:

> To achieve this I used the strategy of striving to simply determine causal relationships. In attempting to determine causes, I found that the internal erosion of soil from under the dam causes broken pipes causing slow drainage resulting in uplift and in turn sliding.

The explanation is constructed by integrating the descriptions of each story in the line of explanation with the appropriate setting, theme and resolution. The final explanation presented to the end-user for our demonstration example is given below.

> We have a concrete dam under an excessive load. I attempted to find the cause of the excessive load. Not knowing the solution and based on the broken pipes in the foundation of the dam, and the downstream sliding of the dam, and the high uplift pressures acting on the dam, and the slow drainage of water from the upstream side of the dam to the downstream side I was able to make an initial hypothesis. To achieve this I used the strategy of striving to simply determine causal relationships. In attempting to determine causes, I found that the internal erosion of soil from under the dam causes broken pipes causing slow drainage resulting in uplift and in turn sliding. This led me to hypothesize that internal erosion was the cause of the excessive load. Feeling confident in this solution, I concluded that the internal erosion of soil from under the dam was the cause of the excessive load.

having the freedom to reorganize the transitions between hypotheses to follow the explanatory knowledge.

> I attempted to find the cause of an excessive load on a concrete dam. Based on the water marks on the abutments and the debris on top of the dam, I made an initial hypothesis. In looking at causal relationships, I found that a flood would cause the water marks and debris. This led me to hypothesize a flood was the problem. However, based on the duration of recent floods I was able to refute this hypothesis. In evaluating the hypothesis, I found no floods of sufficient duration to have caused the observed problems. As floods and settlement often have similar symptoms, I hypothesized that settlement was the problem. After evaluating the hypothesis and determining causing relationships, I was able to further support this hypothesis. In evaluating the hypothesis, I found the drainage and uplift pressures were consistent with settlement as settlement will cause slow drainage in turn causing high uplift pressures. This led me again to hypothesize settlement. However, based on the selective breaking of the broken pipes in the foundation, I was able to refute this hypothesis. Again in looking at causal processes, I noted that settlement would cause crushed-like damage to the drainage pipes whereas erosion of soil would cause selective breaking. Therefore, I concluded erosion was causing the excessive load.

At the other end of the spectrum, an end-user with little expertise in the domain may not be aided at all by an explanation that follows the expert system's possibly diverted line of reasoning [5]. For such an audience, the explanation system can be given complete freedom to find the most direct line of explanation, thus moving the end-user to the solution as directly as possible. By choosing the problem constraint *Direct-Indirect* and the solution constraint *No-RC*, REX finds the most direct line of explanation, as defined by the specification and the case being solved, in the risk analysis domain.

> I attempted to find the cause of an excessive load on a concrete dam. Based on the broken pipes in the foundation, the sliding of the dam, the uplift pressures, and the slow drainage, I was able to make an initial hypothesis. In studying causal relations, I found that the erosion of soil from under the dam would cause broken pipes, resulting in slow drainage, thereby creating increased uplift pressures and eventually sliding of the dam downstream. This led me to conclude erosion was the cause of the excessive load.

Clearly there are audiences that fit between these two extremes. For such an audience, it may be more appropriate to restrict the explanation system to follow only what the expert system did and not give it the ability to introduce new information. By choosing the problem constraint *Direct-Indirect* and the solution constraint *Only-RC*, an explanation can be found that will correspond to the most direct movement through the data uncovered by the expert system to the final conclusion. In the risk analysis example, this results in the following explanation:

> I attempted to find the cause of an excessive load on a concrete dam. Based on slow drainage and high uplift pressures, I made an initial hypothesis. In studying the causal relationships, I found that settlement of the dam would cause the slow drainage which would in turn create high uplift pressures acting on the dam, thereby suggesting settlement as the problem. However, based on the non-uniform damage

> of the broken pipes in the foundation, I was able to refute this hypothesis. Again in looking at causal processes, I noted that settlement would cause crushed-like damage to the drainage pipes whereas erosion of soil would cause the observed selective damage. Therefore, I concluded erosion was causing the excessive load.

These examples show that unlike the traditional approach to expert system explanation, the reconstructive approach has the ability to create a spectrum of coupling between the expert system's line of reasoning and the explanation system's line of explanation. This spectrum can be used to influence the nature of the explanation depending on the general characteristics of the audience.

7. Discussion

This research has shown four major results:

(1) limitations of previous expert system explanation techniques have been identified;
(2) a theory of reconstructive expert system explanation has been defined and shown to provide the potential for greater power and flexibility than other approaches;
(3) the feasibility of using this reconstructive theory has been shown through the REX system; and
(4) examples have been presented of generated explanations which address audience needs not easily accommodated by more traditional approaches.

The two major shortcomings of the reconstructive paradigm are the potential for inconsistencies between the line of reasoning and the line of explanation, and the additional cost of building, maintaining, and searching the explanatory knowledge base. As discussed earlier, we have attempted to use a series of constraints and the audience, goal, and focus of the explanation in order to reduce the ill-effects of the inconsistency problem. Likewise, we have used a high-level specification of the expert system's knowledge base as an interface to the explanation system's knowledge base in order to reduce the dependence between the two. With a reduced dependence comes the potential for reduced maintenance costs. Overall, however, these two problems remain as major foci for future research on the reconstructive paradigm. Included in this investigation will need to be research aimed at further guarantees on the consistency agreement between the line of reasoning and the line of explanation, and more systematic methods of choosing the proper constraints, level of coupling, and explanation viewpoints for classes of expert system users as well as for individual users.

In addition to the above investigations, reconstructive expert system explanation highlights several other areas of future research that may significantly benefit expert systems. For example, as explanation is now viewed as a problem-solving task largely distinct from the expert system's domain problem solving, there exists the potential for feedback from the process of explanation to the process of domain problem solving. In this way, explanation can become an active element of the overall problem-solving process. Active explanation is a well-known phenomenon in human explanation. Nearly everyone has had the experience of being stuck on a problem and while explaining their approach to a colleague, the solution somehow 'pops' into their mind. *Active expert system explanation* is thus the study of how to identify and use the potential feedback that exists from an active, problem-solving explanation system to a problem-solving expert system.

Concurrent reconstructive expert system explanation is another area of future interest. This study has shown several advantages to the use of reconstructive explanation for retrospective queries. Concurrent reconstructive explanation would focus on bringing the implications and advantages of the retrospective paradigm to the more general problem of questions posed during the execution of the expert system. Such an investigation must identify the nature of the problem solving required to answer such concurrent queries. Further, the implications of providing consistent, concurrent explanations must also be investigated. For example, a line of explanation may appear appropriate at one time but after further problem solving by the expert system, that line of explanation may become hopelessly inconsistent with the expert system's behavior.

In the extreme case, in which concurrent explanation queries are frequently asked, it may be necessary and advantageous to adopt an *explanation-based problem solving* approach. Frequent queries to the expert system during problem solving may require that consideration of what reasoning strategies will be understood by the end-user before problem solving takes place. For example, the expert system could consult user models to determine any preferences that the end-user is likely to have for certain reasoning steps. Based on this information, the expert system could actually reason according to the end-user's model of the domain. Thus the nature of the problem solving becomes explanation-based.

Finally, one other interesting area of investigation is suggested by this research. The explanation systems built in our feasibility study have been relatively small and as such, techniques like A* have worked fine. However, as the domain becomes more complicated, the construction of the explanation will begin to require more and more intelligent control. In fact, this control appears to be analogous to the control knowledge of the expert system. Attempting to build a reconstructive explanation system in such a domain would require the construction of an *expert explanation system*. Expert explanation would require capturing the knowledge of how experts go about the process of reconstructing an explanation to their problem solving. The result would be an expert system for constructing the explanation and an expert system for solving the problem. Clearly, many of the issues involved in the research presented here become magnified in such an investigation. However, the notion of an expert explanation system is an appealing thought.

Overall, we have presented a first step in the computational study of reconstructive explanation as a complex problem-solving process. We have demonstrated several advantages to the general paradigm as well as pointed out several significant disadvantages. We believe that many exciting research topics remain open and hope that this report will serve as a spring-board for future investigation.

Acknowledgement

This research was supported by National Science Foundation grant number IRI-8715623.

References

[1] J.S. Brown, R. Burton and J. de Kleer, Pedagogical, natural language and knowledge engineering techniques in SOPHIE I, II, III, in: D. Sleeman and J.S. Brown, eds., *Intelligent Tutoring Systems* (Academic Press, New York, 1982).

[2] B. Chandrasekaran, M.C. Tanner and J.R. Josephson, Explaining control strategies in problem solving, *IEEE Expert* **4** (1) (1989) 9–24.

[3] W.J. Clancey, Heuristic classification, *Artif. Intell.* **27** (1985) 289–350.

[4] R. Cohen, Producing user-specific explanations, in: *Proceedings AAAI Workshop on Explanation* (1988) 44–47.

[5] P.H. Erdman, *A Comparison of Computer Consultation Programs for Primary Care Physicians: Impact of Decision Making Model and Explanation*, Ph.D. Thesis, University of Wisconsin–Madison (1983).

[6] K.A. Ericsson and H.A. Simon, *Protocol Analysis: Verbal Report as Data* (MIT Press, Cambridge, MA, 1984).

[7] K.A. Ericsson and H.A. Simon, Verbal reports as data, *Psychol. Rev.* **87** (1980) 215–251.

[8] E.A. Feigenbaum, The art of artificial intelligence, in: *IJCAI-77*, Cambridge, MA (1977) 1014–1029.

[9] B. Franck, *Preliminary Safety and Risk Assessment for Existing Hydraulic Structures—An Expert Systems Approach*, Ph.D. Thesis, Mechanical Engineering Department, University of Minnesota, Minneapolis (1987).

[10] D.W. Hasling, W.J. Clancey and G. Rennels, Strategic explanation for a diagnostic consultation system, *Int. J. Man-Mach. Stud.* **20** (1984) 3–19.

[11] E. Horvitz, D. Heckerman, B. Nathwani and L. Fagan, The use of a heuristic problem-solving hierarchy to facilitate the explanation of hypothesis-directed reasoning, in: *Proceedings MEDINFO'86* (1986) 27–31.

[12] P. Johnson, I. Zualkernan and S. Garber, Specification of expertise, *Int. J. Man-Mach. Stud.* **26** (1987) 161–181.

[13] A.M. Keuneke, *Machine Understanding of Devices: Causal Explanation of Diagnostic Conclusions*, Ph.D. Thesis, Ohio State University (1989).

[14] J.M. Mandler and N.S. Johnson, Remembrance of things parced: Story structure and recall, *Cogn. Psychol.* **9** (1977) 111–151.

[15] K.R. McKeown and R.A. Weida, Highlighting user related advice, in: *Proceedings AAAI Workshop on Explanation* (1988) 38–42.

[16] K.R. McKeown, M. Wish and K. Matthews, Tailoring explanations for the user, in: *Proceedings IJCAI-85*, Los Angeles, CA (1985) 794–798.

[17] J.D. Moore and W.R. Swartout, A reactive approach to explanation, in: *Proceedings IJCAI-89*, Detroit, MI (1989).

[18] A. Newell, The knowledge level, *AI Mag.* **2** (2) (1981) 1–20.

[19] C.L. Paris, Combining discourse strategies to generate descriptions to users along a naive/expert spectrum, in: *Proceedings IJCAI-87*, Milan, Italy (1987) 626–632.

[20] C.L. Paris, M.R. Wick and W.B. Thompson, The line of reasoning versus the line of explanation, in: *Proceedings AAAI Workshop on Explanation* (1988) 4–7.

[21] J.P. Ryan and S. Bridges, Constructing explanations from conceptual graphs, in: *Proceedings Third Annual Workshop on Conceptual Graphs* (1988) p. 4.12.

[22] R.D. Schachter and D. Heckerman, Thinking backward for knowledge acquisition, *AI Mag.* **8** (3) (1987) 55–61.

[23] E.H. Shortliffe, *Computer-Based Medical Consultations: MYCIN* (Elsevier, New York, 1976).

[24] W.R. Swartout, XPLAIN: a system for creating and explaining expert consulting programs, *Artif. Intell.* **21** (1983) 285–325.

[25] P.W. Thorndyke, Cognitive structures in comprehension and memory of narrative discourse, *Cogn. Psychol.* **9** (1977) 77–110.

[26] D.D. Tukey, *A Psychological Study of Subject's Methods of Conducting Experiments in a Rule-Learning Task*, Ph.D. Thesis, University of Minnesota (1983).

[27] J.W. Wallis and E.H. Shortliffe, Explanatory power for medical systems; Studies in the representation of causal relationships for clinical consultations, *Methods Inf. Med.* **21** (1982) 127–136.

[28] J.L. Weiner, BLAH, a system which explains its reasoning, *Artif. Intell.* **15** (1980) 19–48.

[29] M.R. Wick, *Reconstructive Explanation For Expert Systems*, Ph.D. Thesis, University of Minnesota (1989).

[30] M.R. Wick and J.R. Slagle, An explanation facility for today's expert systems, *IEEE Expert* **4** (1) (1989) 26–36.

[31] F. Wilson, *Explanation, Causation and Deduction* (Reidel, Dordrecht, Netherlands, 1985).