An Architecture for Rule Based System Explanation

T. R. Fennel
J. D. Johannes
University of Alabama in Hunstville
Huntsville, AL 35899

Abstract

This paper presents a system architecture which incorporate both graphics and text into explanations provided by rule based expert systems. This architecture facilitates explanation of the knowledge base content, the control strategies employed by the system, and the conclusions made by the system. The suggested approach combines hypermedia and inference engine capabilities. Advantages include: closer integration of user interface, explanation system, and knowledge base; the ability to embed links to deeper knowledge underlying the compiled knowledge used in the knowledge base; and allowing for more direct control of explanation depth and duration by the user. User models are suggested to control the type, amount, and order of information presented.

Introduction

One of the earliest claims of expert system developers was that the resulting systems could "explain" their actions. These claims were often effectively backed up by the textual presentation of traces of rule firings which could explain "how" the system had made a decision.[Pople, '77] Additionally, systems could answer "why" the system was asking for information by presenting as explanation an English text description of the rule which required the information [Clancey, '83]. However, complete explanation requires addressing the problems of what , how, when and to whom knowledge is to be communicated.

[Wick and Slagle, '89] suggest that explanation capabilities could be greatly enhanced by the introduction of supplementary knowledge and by allowing variations of queries over time. For example, the user could ask not only "Why do you want to know this now?", but could also ask "Why would you ever ask me for this information?". Similarly the user could ask not only "How did you know?" , but also "How could you find out?". To answer these questions the system must keep extended histories, or traces, of actions taken by the expert system and based on dependencies be able to generate responses of a forward looking nature.

[Chandrasekaran, Tanner, and Josephson, '89] emphasize that explanation should be provided not only at the low levels (exemplified by presenting the conditions associated with a single specific rule) but that high-level explanation of overall system goals should also be available. Their suggestions are supported by work on automatic generation of textual explanations through specialized grammars [Bridges and Johannes, '89] . An underlying truth here is that humans tend to be much better at explaining their actions because they are able to convey both their abstract goals and detailed information -- but with the significance of the details "slanted" towards satisfying the stated goals. Therefore, the grammar used by humans during explanation goes beyond that used for simply explaining system details.

It has been suggested that much of the difficulty in developing expert systems is that the early recorded sessions provide better explanation knowledge than actual problem solving knowledge. To obtain better explanations, the overall system must be designed and developed with explanation as an integral part in all project stages. Particularly for rule based systems we advocate working with a representation of "IF, THEN, BECAUSE" rules as opposed

to "IF, THEN" rules. By giving early knowledge aquistion sessions very heavy weight, useful explanation knowledge can be gathered. Subsequently, this explanation knowledge can be used for verification of solutions provided by the expert system.

A final point needs to be made in this introduction. There is a difference between "justification" and "explantion". For justification we must provide a formal proof of completeness. For explanation, we must focus on and explain only the part of a solution that was not understood. [Clancey, '83]

An Architecture for Explanation

Figure 1 presents a knowledge based system architecture which emphasizes explanation capabilities on an equal par with capabilities provided to solve the original problem. The three main components of the architecture are an expert system intended to solve the "original" problem, an explanation system, and a user interface which acts as a bridge between the two. We will now discuss these three major components and suggest finer grained architectures for each.

The Expert System

The expert system architecture presents a bottom-up separation of four main layers for conventional data structures, object definitions, flat domain knowledge rules, and meta (or control) rules. This is the system proposed to solve the original task. Imposing the defined structure aids in system development, delivery and maintenance. For example, it aids in development (particularly for larger systems) by allowing for clear delineation of tasks among multiple members of a development team and by making interfaces explicit. The imposed structure aids in delivery by forcing incorporation of conventional database technology (and therefore exploits existing databases) at the lowest level and by separating the interface layer for early attention. By including the general architecture in the support documentation, maintenance tasks requiring different skills can be anticipated.

The lowest level of the expert system represents an underlying database with basic facts about the problem or about the current state of the world as the knowledge base knows it. At the next highest level, an object hierarchy is provided and the object definitions are all linked to conceptual definitions. The values for object attributes are updated through a link with the underlying database. A common mistake in the past has been to assume that the object layer was the lowest layer required for expert systems. This mistake delayed the integration of many systems with conventional databases and often resulted in significant efforts to rewrite the system or duplication of data. The third layer represents a relatively flat body of rules which typically represent a hierarchy of symptoms or contraints and are the result of knowledge engineering. Strategies for applying the constraint rules are represented in the fourth level by another hierarchy of meta-rules (these typically control the inferencing process at a very high level and depend on the built in mechanisms of forward or backward propogation for results at the constraint level).

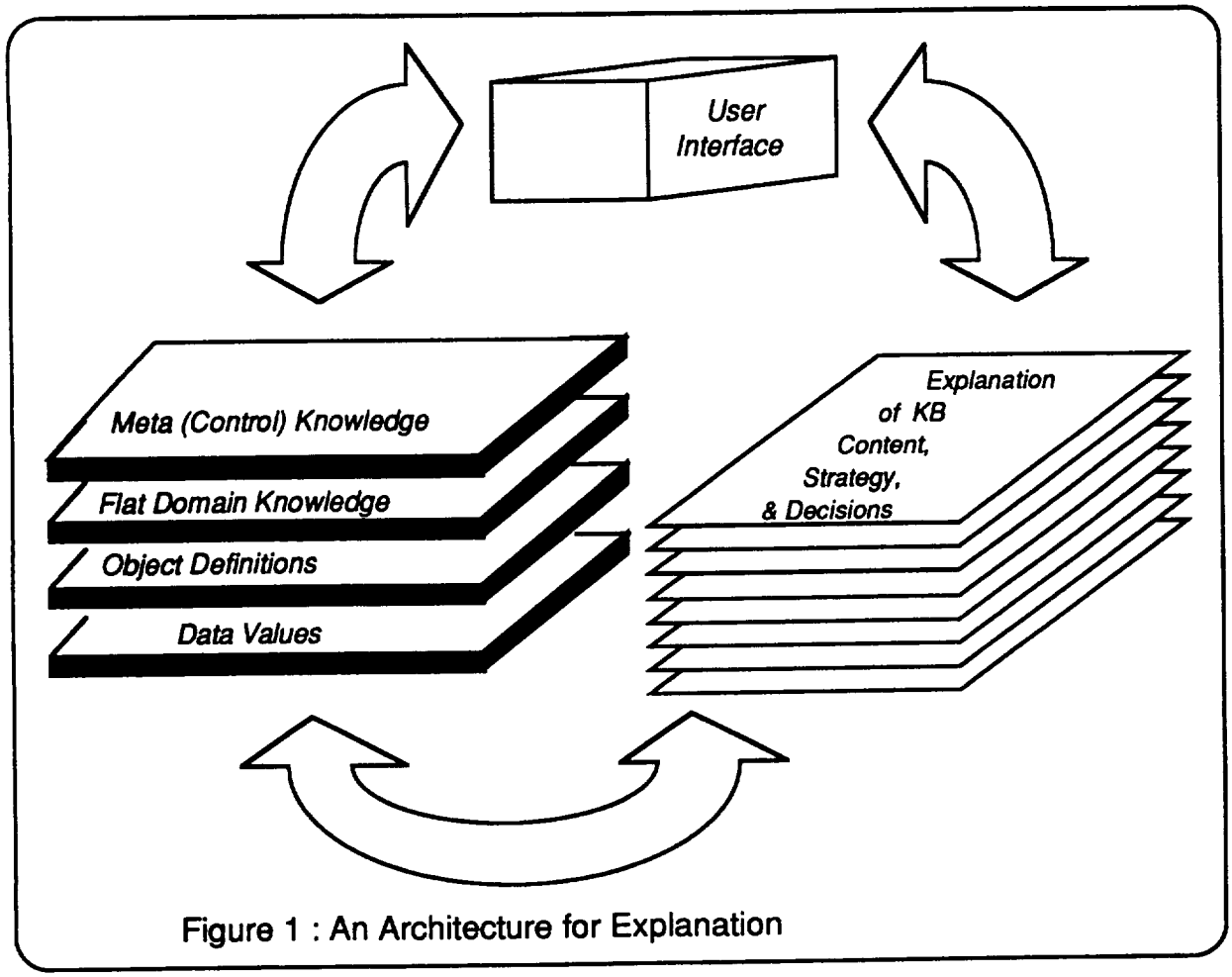

Figure 1 : An Architecture for Explanation

It turns out that imposing a structure upon the expert system can help provide structure to explanations. For example, more sophisticated implementations can use this imposed structure along with knowledge about each layer and the user to guide planning for explanation construction.

The Explanation System

The second major component of the architecture is the explanation system. Although our architecture would facilitate such an implementation, we do not insist that the portion of the system which provides explanation be a "complete" separate expert system. That is, we do not insist that the explanation system be capable of solving the original problem, as does [Wick and Thompson, '89].

At the lower levels in the architecture, the explanation system layers correspond to those in the expert system. For example, the bottom layer consists primarily of simple information about the data values layer of the expert system and contains typical information typical of that found in a "data dictionary", such as data value types, ranges, sources, etc. The next layer corresponds to the object definition layer in the expert system. It contains information about the objects and knowledge about fundamental mechanisms such as inheritance through the class structures of the implementation system. At the next two levels (the flat domain knowledge and control knowledge layers) the explanation system can contain even more information than the corresponding expert system

layers. The internal architecture for these layers is dependent upon the depth of background material included, the representation used for depicting strategy, and the extent of any user models.

To understand how the architecture may change it is important to specify the type of explanation is to be supported. In the following three sections we present approaches working within the depicted architecture for explaining knowledge base content, strategies, and decisions. [Chandrasekarn, et al, '89] provide details regarding this three pronged approach for explanation from introspection of knowledge and inference.

Explaining Knowledge Base Content

The architecture presented provides for explanation of knowledge base content at all levels. Starting with the lowest level, an underlying database represents basic facts about the problem or about the current state of the world as the knowledge base knows it. Explanation content for the database is typical in that it should provide information on the data sources, last update, units of measure, and validity intervals.

At the next highest level, an object hierarchy is provided and the object definitions are all linked to conceptual definitions. Graphics depicting component and subcomponent details are used where appropriate. Information provided about each object class include its importance in the problem to be solved and how it is used in the problem solving process. Each object attribute is similarly treated with the addition that each object attribute is also flagged to indicate whether its value is simply read in from the database or can be changed by the problem dynamics. The idea of assigning values of LABDATA to data that typically requires no explanation other than source was suggested in [Davis and Buchanan, '77]. Where attributes can have multiple values, the meaning of the multiple values is explained, along with expected consequences on the problem solving process.

The constraint rules form the third level of the knowledge base and serve to emphasize that in a rule based system oriented towards explanation the rules themselves should be thought of as objects. For explanation of content, one successful implementation uses an "index" that graphically shows the constraint hierarchy as composed of only keyword phrases. Additionally, each rule should be captured in hypertext form, so that the user can select any rule from the keyword hierarchy, then any part of the rule can be selected to explain the contents in more detail. Rule attributes include static English text which restates the rule, the rule originator, last update, a list of pointers to any related "cases" or "tests" from which the rule was derived, the relation to other rules, an understandable English text prompt used in conjunction with the rule when requesting information, and a graphical representation of the rule where possible. For systems which use confidence factors, it is imperative to note that the confidence factors themselves convey knowledge that should be explained.[Davis and Buchanan, '77] A confidence factor of one indicates that a "shallow" explanation may suffice since the rule is most likely definitional in nature, while confidence factors not equal to one represent the application of judgement and the relevant ranking of its importance and therefore requires more explanation.

Explaining the Knowledge Based System Strategy

Explaining a strategy involves in part the explaining of a perspective for relating rules hierarchically and then showing how these relations provide leverage for managing a large amount of data or number of hypotheses. The meta-rules at the fourth level of the expert system form the core of solution strategy and can also represented in the explanation system by a graphic hierarchy. At this level the source for the rules becomes critical as these are the rules which control the order for checking the constraints at the next lower level. These rules explicitly determine which constraints are checked under varying circumstances. The strategies implemented intentionally mimic those used by experts from various areas within the domain and are one of the areas where having multiple

experts can be an advantage and where explaining the source of a rule can make a significant difference in acceptance of the final system.

It is clear that strategy and structure are intimately related in the expert system. The explanation system should make this as explicit as possible. For example, where screening clauses are used in the system for internal control as part of a rule, the rationale behind their placement should be documented and available for explanation.

Explaining Knowledge Based System Decisions

The ideal situation when explaining decisions is to employ any material a person may use, the point being to represent the "bottom line" as clearly as possible. For example, the rule hierarchies presented to explain the knowledge base content can be enhanced by highlighting information (the computers equivalent of pointing) used in the decision process. Where graphical keyword "indices" are available, they can be used to highlight a single keyword representing a rule or group of rules while presenting the constraint hierarchy. This will often serve as sufficient explanation for domain experts, while hypermedia links from the keyword hierarchy provide the "back pocket" type of information needed for explanation to other audiences.

Following the example set by [Brown, Burton, and de Kleer, '82], system developers should attempt to anticipate what are most likely to be the more difficult areas involved in making the decisions and provide even more depth and tutorial information for explanation of decisions in some areas. The areas can be highlighted by assigning individual measures of complexity or importance to individual rules and viewing the hierarchy of rules with aggregate weighting above a threshold value.

Even artificially constructed "trees" representing the HOW information for decision explanations can be very useful. That is, the tree presented as explanation need not reflect the reasoning process in it's entirety (Wick and Thompson's work argues that it may not reflect it at all!). However, we feel that it should demonstrate at least a "feeling" for the structure of the problem space and the nature of the search strategy used.

One of the more difficult tasks may be tying explanations to a general abstract level or task (such as in Dr. C's work), especially for strategy rules. Developers of explanation systems must realize that any rule, no matter how obvious or clear, is only a single step in the explanation. Other steps include setting strategy context (such as generic task identification), focusing on state information (particular values of object attributes at a point in time), and elucidation of outcome.

The User Interface

The user interface is the third major component of our suggested architecture and spans the gap between the expert system and the explanation system. In the past, most expert systems have typically relied on simple menu driven interfaces and textual presentation of explanations. Notable exceptions to this include the STEAMER [Hollan, Hutchins, and Weitzman, '84] system which used an underlying simulation model with incorporated graphics and the General Electric DELTA expert system for diagnosing diesel electric locomotive failures which incorporated video storage as part of the system [Bonissone and Johnson, '83]. More recent work by [Sue, '89] has focused on a mixture of text and graphics in the explanations.

We suggest that the interface should be some implementation of hypermedia. The recent emergence of robust hypermedia systems argues favorably for the integration of graphics and text. An ancient Chinese proverbs states "It is better to see a thing once than to read about it one hundred times." The wisdom of this statement has been proven repeatedly by people who while trying to explain their actions to others resort to the use of a graphic as part of their clarification [Berry and Broadbent, '87]. Therefore, perhaps the best rationale for incorporating graphics and text is simply to mimic reliance upon them as

humans do. By using figures which have been scanned in, and then adding "buttons" or links to additional information and text fields we can allow for perusal of a tremendous amount of information at a level dynamically controlled by the user. It is important to realize that the links created for explanations tend to be more specific than those created for a purely informative stack -- at least at the beginning of the explanation. However, as the user traverses links away from the starting point the bounds on what type of information is presented is left up to the system developers (for example, it may be desirable to restrict the amount of autonomy afforded to students where the explanation system also serves as part of an intelligent tutoring system).

Modern portable computers, optical discs, and graphics software make it possible to quickly and easily capture and integrate graphic material. The architecture suggested combines database, hypermedia and inference engine capabilities. These capabilities are readily available on conventional PC hardware and recent announcements in the area of integration across diverse packages makes it practical to expect easier access to such tools.

User Models and Explanation in Intelligent Tutoring Systems

An additional level of complexity is added to the problem of explanation when we introduce the need for models of the user so that the information presented will be both understandable and timely. Related work [Wenger, '87] in the rapidly expanding field of intelligent tutoring systems demonstrates repeatedly that it is the communication of knowledge (not just data) that is important and that the presenter of knowledge must make allowances for student abilities. Most explanations are presented to a single individual, or at least to a group with focused attention in a common setting. For example an expert system developed as an engineering aid may be used repeatedly by individual engineers who are experts in the domain. However; when explaining the actions of the system (which have led to specific decisions) during a formal review, the experts must be able to integrate background information, current focused information, and their overall goals into explanations at a level their audience will understand. The point is that the same explanations given by the system to the expert during its normal use will not suffice as explanations given to a broader audience. The task of trying to model even the typical user (in an effort to know what to present and how to present it) is often not straightforward.

We would like to continue to investigate the use of expert systems as intelligent tutors. Conceptual definitions of objects and rule hierarchies are used extensively in explanations, and serve as excellent starting places for those using the system as a tutor. These hierarchies can be used for quickly identifying areas of interest to different users and for providing a type of dialogue from which students can ask for more detailed explanations [Moore and Swartout, '89].

Any good explanation must "make contact" with previously known concepts. It's a good idea to include a core of short tutorials on the fundamental concepts in an explanation system. These provide the needed basics upon which everyone can view the explanations. They also provide a good example of what should not be included in the system used to solve the problem, but should be included in the explanation system.

Capturing the "Link" between Compiled and Deep Knowledge

The deliberate separation of the explanation system component from the expert system highlights the fact that knowledge required for explanation often lies outside of the knowledge incorporated into the expert system designed to solve the original problem. One basic reason for this is that the expert system represents compiled knowledge. The idea here is similar to that found in conventional compiled languages where efficiency is gained by stripping away non-executable code such as comments. In several ways, explanation knowledge is very similar to the knowledge often gathered into documentation for conventional programs. Just as conventional program documentation has many levels , the explanation system is a knowledge based system with multiple layers (although not always true in practice, there should be major differences between a user guide and detailed programmers guide to the same software).

It is the high level and abstract knowledge (such as originally intended use, goals, or even current events such as budgetary constraints) that is often compiled out of the final version of a knowledge base. As a result, explanations associated with expert system will most likely be later questioned regarding completeness, accuracy, or accountability -- and the true explanations may not be available. We've found that the most difficult part of this is indeed deciding how to tie explanations to the higher level goals. In many cases we recommend simple English text statements as they seem most appropriate. The more abstract problem solving goals (such as the strategy rules) are depicted using process flow diagrams. A fairly simple mapping allows for capturing the link between the strategy rules and the rules at lower layers.

Future Directions

It has been suggested [Brown, Burton, and de Kleer, '82] that links to conceptually faithful simulations can provide for a form of continuous explanations and could thereby represent a deeper knowledge of the domain. We would like to pursue this area by providing links from the hypermedia interface to an application written for simulating processes in the domain.

Construction of an appropriate grammar for describing the relationships among objects and rules within the domain and specialized for use in explanations is being considered for future research. The grammar definition would help ensure future applications would find the embodied knowledge in machine intelligible formats and could be used to limit the scope of explanations which must be generated. It has been suggested [Paris, '88] that explanation for expert systems provides a rich domain in which to study natural language generation. The simple architecture presented in this paper would be refined to accomodate a more intelligent architecture for the explanations system.

Summary

An architecture has been suggesteed for combining an expert system, explanation system and hypermedia based user interface. Components of explanation include explaining knowledge base content, strategy, and decisions. By emphasizing explanation as a major system goal which requires knowledge and effort aside from solving the original problem, the systems can benefit : by being more readily received in the end user environment; by also serving as a beginning platform for instruction; by providing links to the deeper knowledge underlying that which would normally be compiled out of the knowledge base; and by providing for smoother integration of interface, knowledge base, and data which helps ensure they will continue to be used.

References

Abu-Hakima, S., A Generic Environment for Constructing Diagnostic Hierarchies,in *Proceedings of the IJCAI-89 Workshop on Integrated Human-Machine Intelligence in Aerospace Systems*, pp. 90-99, 1989.

Berry, D.C., and Broadbent, D.E., Expert Systems and the man-machine interface. Part Two: The user interface, *Expert Systems,* Vol. 4, No. 1, February 1987.

Bonissone, P.P. and Johnson, H.E., Expert system for diesel electric locomotive repair, *Knowledge-based Systems Report,* General Electric Co., Schenectady, N.Y., 1983.

Bridges, S., and Johannes, J.D., Integration of Knowledge Sources for Explanation Production, *1989 ACM,* 1989.

Brown, J.S.; Burton, R.R.; and de Kleer, J., Peagogical, Natural Language and Knowledge Engineering Techniques in Sophie I, II, and III, in I*ntelligent Tutoring Systems*, D. Sleeman and J.S. Brown, eds., Academic Press, London, UK, pp 227-282, 1982.

Chandrasekaran, B.; Tanner, M.C.; and Josephson, J.R., Explaining Control Strategies in Problem Solving, *IEEE Expert,*, pp. 9-24, Spring 1989.

Clancey, W.J., The Epistemology of a Rule-Based Expert system -- A Framework for Explanation, *Artificial Intelligence*, pp. 215-251, May 1983.

Davis, R. and Buchanan, B., Production Rules as a Representation for a Knowledge-based Consulation Program, *Readings in Knowledge Representation*, Brachman and Levesque eds, 1977.

Hollan, J.D.; Hutchins, E.L.; and Weitzman, L. STEAMER: an interactive inspectable simulation-based training system, *AI Magazine*, vol. 5, no. 2, pp. 15-27, 1984.

Moore, J.D., and Swartout, W. R., A Reactive Approach to Explanation, presented at the AAAI Workshop on Explanations, 1988.

Paris, C.L., Generation and Explanation: Building an explanation facility for the Explainable Expert Systems framework, submitted to the 1988 Workshop on Text Generation, 1988.

Pople, H.E., The Formation of Composite Hypostheses in Diagnostic Problem Solving, in *Proc. Fifth IJCAI*, Morgan Kaufmann, Los Altos, Calif., pp. 1030-1037, 1977.

Weld, D.S. , Explaining complex engineered devices. *BBN Report 5489.* Bolt Beranek and Newman Inc., Cambridge, Massachusetts, 1983.

Wenger, E., *Artificial Intelligence and Tutoring Systems* , Morgan Kaufmann Publishers, Inc., 1987.

Wick, M.R. and Slagle, J.R., An Explanation Facility for Today's Expert Systems, *IEEE Expert*, pp. 26-36, Spring 1989.

Wick and Thompson, Reconstructive Explanation: Explanation as Comlex Problem Solving, in *Proc. Eleventh IJCAI*, Morgan Kaufmann, Los Altos, Calif., pp. 135-147, 1989.