N90-27299

Graphical Explanation in an Expert System for Space Station Freedom Rack Integration

F.G. Craig, D.E. Cutts, & T.R. Fennel
Boeing Computer Services M/S JA-74
Huntsville Artificial Intelligence Center

B. Purves
Boeing Aerospace M/S JA-62
499 Boeing Blvd.
Huntsville, AL 35824-6402

Abstract

This paper focuses on the rationale and methodology used to incorporate graphics into explanations provided by an expert system for Space Station Freedom rack integration. The rack integration task is typical of a class of constraint satisfaction problems for large programs where expertise from several areas is required. Graphically oriented approaches are used to explain the conclusions made by the system, the knowledge base content, and even at more abstract levels the control strategies employed by the system. The implemented architecture combines hypermedia and inference engine capabilities. The advantages of this architecture 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. The graphical techniques employed range from simple static presentation of schematics to dynamic creation of a series of pictures presented "motion picture" style. User models control the type, amount, and order of information presented.

Introduction

The Space Station Freedom (SSF) Program is a complex task requiring the integrated skills of thousands of people. There are many examples within the program of tasks which require the cooperation and participation of several organizations to make critical decisions. As automated expert systems are developed to aid in these decisions and to capture the knowledge from several areas, we should be able to ask them for explanation/justification of their results as we would human experts. The task of rack integration is exemplary of tasks for which justification is required. The racks aboard SSF provide a common element around which design, operational, manufacturing, and logistics decisions are made. The basic task is to decide where racks of a given type should be located aboard SSF. There are several types of constraints which influence the final decision, ranging from operational (such as noisy racks should not be located near crew sleeping quarters) to physical constraints dependent upon other design decisions (such as the general rule that data management system racks, although shielded, should not be unnecessarily located next to potential sources of electromagnetic interference).

The expert system to aid in the integration of this task is documented in detail in an accompanying paper at this conference [1]. This paper will focus on the benefits, methodology, and some of the issues researched for making such systems more usable and complete in the area of explanation.

Explanation

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.[2] 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.[3] However, complete explanation requires addressing the problems of what , how, when and to whom knowledge is to be communicated. In the past, most expert systems have typically relied on textual presentation. Notable exceptions to this include the STEAMER [4] 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[5].

Wick and Slagle [6] 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.

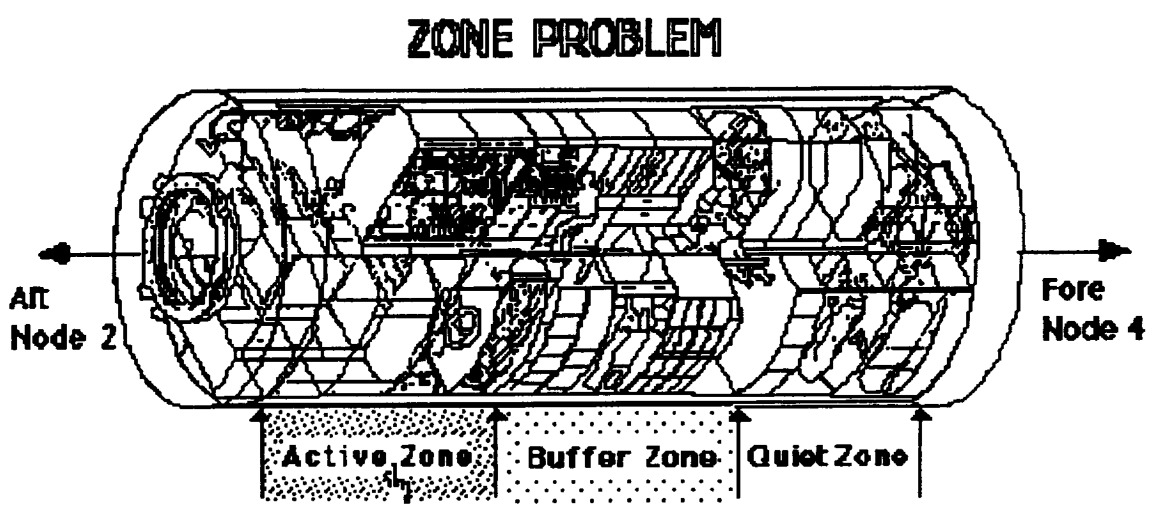
Chandrasekaran, Tanner, and Josephson [7] emphasize that explanation should be provided not only at the low levels (exemplified by presenting the conditions associated with a single specific rule) but that high-level explanation of overall system goals should also be available. Their suggestions are supported by work on automatic generation of textual explanations through specialized grammars [8] . 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.

Most explanations are presented to a single individual, or at least to a group with focused attention in a common setting. 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 [9] 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. 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. (And insistence on understanding is something formal review boards are well known for!) 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.

Rationale for Incorporating Graphics

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 for clarification. For the rack integration task we developed guidelines which dictated that even the quick sketches of an expert should be included as part of the documentation for any rules developed as a result of a knowledge engineering session. Therefore, perhaps the best rationale for incorporating graphics is simply to mimic reliance upon them as humans do.

A sequence of pictures is often very effective at presenting information as it changes over time, and in many situations an appreciable amount of information can be conveyed by a single picture. For example, figure 1 graphically illustrates the noise constraint mentioned previously for the Habitation module of the Space Station. From the figure it becomes obvious that "noisy" racks should be located at the aft end of the module , with a buffer zone located between them and those at the opposite quiet end. Closer scrutiny of more detailed drawings reveals that most of the subsystem related racks such as the Air Revitalization System (ARS) and Thermal Control System (TCS) are located at the noisy end of the module because of the mechanically oriented nature of those racks. However, the galley/wardroom racks are also at the noisy end, indicating that noise associated with the use of a rack is also enough reason to isolate it from the crew sleeping quarters.



The Habitation Module is divided into three zones. The Active Zone supports crew activity and noisy racks. The Buffer Zone has less activity and the racks are not as noisy. The Quiet Zone contains crew quarters, and for the comfort of the crew has little activity and noise.

Figure 1: Noise is a constraint in the Habitation Module

Modern portable computers, optical discs, and graphics software make it possible to quickly and easily capture and integrate graphic material. The architecture chosen for our research combines database, hypermedia and inference engine capabilities. The specific software packages used in the initial effort are Microsoft Excel, Neuron Data's Nexpert Object, and Apple Computer's Hypercard. The hardware platform chosen was a Macintosh II with 8 MB of memory.

Additionally, the recent emergence of very affordable hypermedia systems was also a major contributor in our decision to incorporate graphics. By using figures which have been scanned in, and then adding "buttons" or links to additional information 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 simply for an informational 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 the rack integration expert system we "flavored" the entire network of information by relating to a hierarchy of interface, control, constraints, and state information (the layered approach used here is typical of constraint satisfaction problems we have dealt with in the past and is documented in detail in [1]).

In the following three sections we present our research applied to the three areas of explaining knowledge base content, strategies, and decisions. Chandrasekarn, et al, [7] provide details regarding this three pronged approach for explanation from introspection of knowledge and inference.

Explaining Knowledge Base Content

For our research purposes we have pursued providing explanation of knowledge base content at all levels. Starting with the lowest level, the underlying database represents basic facts about the problem (such as the number of possible locations for racks in each module) or about the current state of the world as the knowledge base knows it (engineers often start their analysis from a baseline configuration of rack assignments and attempt perturbations). For the database we 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 rack integration task 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 by Davis, et al in [10]. Where attributes can have multiple values, the meaning of the multiple values is explained, along with expected consequences on the problem solving process. For example, the "RACK" class which represents the rack objects has an attribute "noise_level_environment_required". The values for this attribute are "sensitive", "not very sensitive", or "not sensitive at all". The effect is that racks which are "sensitive" to noise can only be located in the quiet zone of the Habitation module (noise is not a concern in the Laboratory or Logistics modules).

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. A graphical depiction of the constraint hierarchy is presented using only keyword phrases. Additionally each rule is 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.

Although our current system does not use confidence factors, it is interesting to note that the confidence factors themselves convey knowledge that should be explained.[10] A confidence factor of unity indicates that a "shallow" explanation may suffice since the rule is most likely definitional in nature, while confidence factors not equal to unity represent the application of judgement and the relevant ranking of its importance and therefore requires more explanation.

Explaining the Knowledge Based System Strategy

The control rules at the fourth level of the knowledge base are also represented in 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. Not all constraints are checked for the varying types of racks. For example constraints associated with zoning restrictions based on the type of science are only checked for racks in the Laboratory module. The strategies implemented intentionally mimic those used by experts from various areas within the rack integration domain. For example, the strategy for checking the constraints associated with moving a Laboratory payload rack were derived from knowledge engineering sessions with a payload integration specialist. Because payloads are typically unique, they have widely varying utility requirements. This is exactly one of the areas checked first and is responsible for most problems with integrating payload racks. Justification of this strategy is supported with a graphic depicting the low percentage of common interface plates in the Laboratory module due to payload unique requirements.

Graphically representing generic tasks such as "hierarchical classification" or "plan selection and refinement" has proven to be a very difficult task. Current efforts are focusing on the use of simple conceptual sketches or icons presented in a cyclic manner to emphasize the ongoing and dynamic nature of such tasks.

Explaining Knowledge Based System Decisions

The ideal situation here 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. For the rack placement expert system we incorporated the ability to highlight a single keyword representing a rule or group of rules while presenting results from an analysis (see figure 2). This often served as sufficient explanation for the domain experts, while links from the keyword hierarchy provided the "back pocket" type of information (previously shown in figure 1) needed for justification to other audiences.

Following the example set by [11], we have attempted to anticipate what are most likely to be the more difficult areas involved in making the decisions and have provided even more depth and tutorial type of information for explanation of decisions in some areas. For example, the routing of utilities required by a rack is an area where many of the verification test cases showed that the human experts had the hardest time explaining their actions. For this reason, the assumptions and formula used for calculating weight, volume, and length information for utilities are all well documented and incorporated in explaining decisions affected by routing criteria.

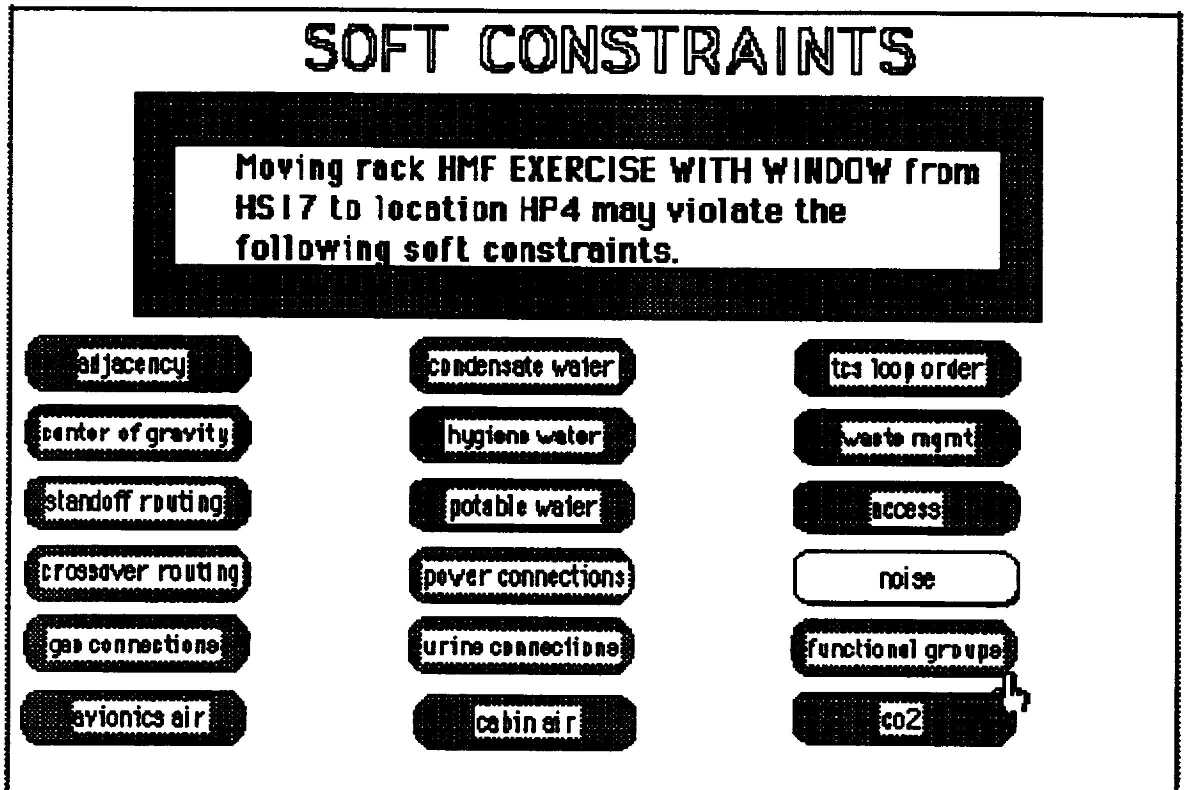


Figure 2: The noise constraint is readily identified

It is for use in explaining decisions that we are developing user models to control the first level of explanation presented to the user. The interface presents only a CAN or CAN NOT decision regarding placement of a rack in a given area and a brief explanation of WHY NOT if the placement was disallowed. Whenever the user asks for further explanation, a "novice" user is presented with a more detailed explanation of the type of problems encountered. An "experienced" user is linked directly to the constraint keyword hierarchy. At the present time, the explanation information presented is mostly static -- prepared beforehand. One of our areas of interest in extending the system is in dynamic creation of explanation objects which would change with the circumstances associated with the knowledge base and with the user. We have made a first step in this direction with the constraint keyword highlighting mechanism mentioned above.

Capturing the "Link" between Compiled and Deep Knowledge

An admission on our part and hopefully a lesson for others is that our first pass at using graphics to explain the knowledge base content was woefully inadequate. It was only when a new member joined our team who was totally unfamiliar with the SSF program that we came to realize this fact. Without knowing it we had been unintentionally "compiling out" knowledge by not representing what we had come to believe was "common sense". For example, we had neglected to document the reasoning behind not allowing racks requiring windows to be placed on the wall facing forward in the SSF orbit. These walls are more subject to meteor hits than the other walls and since windows are regarded as built in safety hazards anyway, they should be located where they are not likely to get hit. Obvious. Right. Another example is where different walls (the Starboard and Floor walls) use the same physical area for routing of utilities. This imposes an additional level of constraints to be checked to satisfy the requirement for separation of redundant systems as illustrated in figure 3.

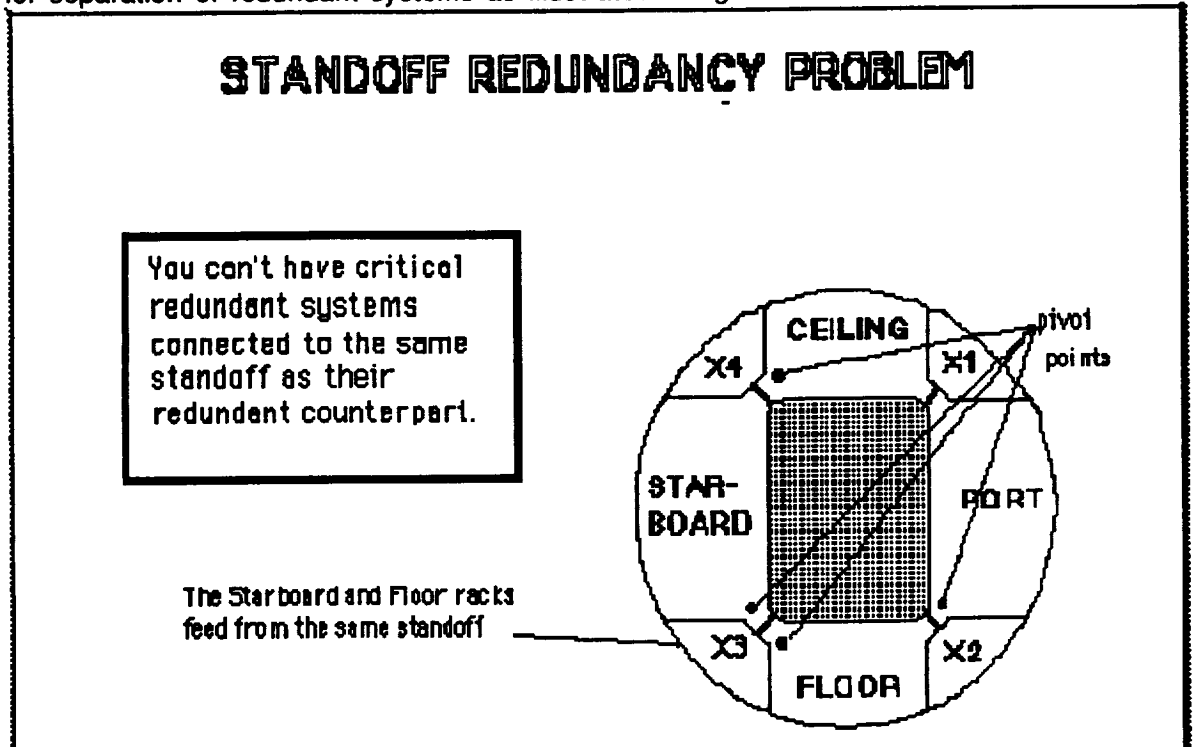


Figure 3: Checking for redundancy constraints requires deeper knowledge of utility routing procedures

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. For the rack integration expert system we have used graphically oriented techniques to document the source, intent, and actual meaning of the knowledge in the knowledge base. We've found that the most difficult part of this is indeed deciding how to graphically represent the higher level goals and in many cases we use simple English text statements as they seem most appropriate. The more abstract problem solving goals (such as the control rules) are depicted using process flow diagrams. A fairly simple mapping allows for capturing the link between the control rules and the constraint rules.

Future Directions

The Apollo program provides proof that much of the data, information, and knowledge associated with large aerospace programs can be lost to later generations. One of the goals of the Space Station Freedom (SSF) program is to ensure that not only is basic data and information available for future access, but also that knowledge available now is also captured for later use by the program. However, while documentation for data or computer programs often have very specific standards imposed upon them, the standards for documentation associated with captured knowledge is still in the formative stages [12]. One of our research goals is to investigate ways of testing how to document captured knowledge. It is fairly easy to understand that just as comment statements form an important part of computer program documentation, explanation capabilities can be used to determine how well a knowledge based system is documented.

We also recognize a need to expand the explanations of why a rack WAS allowed in a given location, not just WHY NOT. The current approach uses the how capabilities of the expert system shell to graphically demonstrate that the control rules were invoked and which constraints were checked.

It has been suggested [13] 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 to an application written for simulating the effects of different routing strategies.

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 was machine intelligible and could be used to limit the scope of explanations which must be generated.

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.

Summary

This research has focused on incorporation of graphics into explanations for a knowledge based system. The test domain chosen was that of rack integration for the Space Station Freedom. This test domain is typical of a class of constraint satisfaction problems and demonstrates that configuration tasks are particularly amenable to effective use of graphics in explanations. Components of explanation include explaining knowledge base content, strategy, and decisions.

By emphasizing explanation as a major system goal 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

1) Craig, F., Cutts, D, and Fennel, T. , A Knowledge-Based approach to Configuration Layout, Justification and Documentation, 5th Conference on Artificial Intelligence for Space Applications, Huntsville, AL, May 1990.

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

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

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

5) 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.

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

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

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

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

10) Davis, R. and Buchanan, B., Production Rules as a Representation for a Knowledge-based Consulation Program, *Readings in Knowledge Representation*, Brachman and Levesque eds, where, when, pages?.

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

12) Carnes, J.R.; Fennel, T.R.; Rice, R.; Trammel, M., Automation and Robotics Plan, *Space Station Freedom Data Requirement SE08*, draft 1988

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