

ORIGINAL PAGE IS  
OF POOR QUALITY

## BROWSING SCHEMATICS: QUERY-FILTERED GRAPHS WITH CONTEXT NODES

Eugene C. Ciccarelli  
Bonnie A. Nardi

IntelliCorp  
1975 El Camino Real West  
Mountain View, California 94040

### Abstract

This paper reports the early results of a research project to create tools for building interfaces to intelligent systems on the NASA Space Station. One such tool is the Schematic Browser which helps users engaged in engineering problem solving find and select schematics from among a large set. Users query for schematics with certain components, and the Schematic Browser presents a graph whose nodes represent the schematics with those components. The query greatly reduces the number of choices presented to the user, filtering the graph to a manageable size. Users can reformulate and refine the query serially until they locate the schematics of interest. To help users maintain orientation as they navigate a large body of data, the graph also includes nodes which are not matches but provide global and local context for the matching nodes. Context nodes include landmarks, ancestors, siblings, children and previous matches.

### INTRODUCTION

As part of the Space Station effort, The National Aeronautics and Space Administration (NASA) is sponsoring several demonstration projects to show the utility of intelligent systems built with knowledge-based software technology. The Space Station will be an extremely complex system composed of many subsystems, each with considerable complexity of its own. Managing the complexity inherent in the structure and behavior of the engineered systems is a major challenge. The use of intelligent systems is intended to help manage that complexity by automating some of the functionality that is currently handled for analogous systems in a manual mode. But it will be critical that humans be able to understand, interact with, and control the intelligent system when necessary. To that end, suitable human-computer interfaces must be built.

Our project, funded by the NASA Johnson Space Center, is to build some tools for the

construction of those interfaces. This paper reports the early results of our efforts to build one such tool, a graphic browser for retrieving schematics (engineering diagrams). The general context of use for the browser in our project is a diagnostic task, although it would be useful for other tasks such as training and maintenance. End users of the interfaces built with the tools are flight control engineers, test engineers and astronauts. Users of the tools are knowledge system developers. We are using the thermal bus (heat transport system) of the Space Station as an example of an engineered system for demonstrating our work.

The use of schematics is central to engineering problem solving. Schematics show connectivity between components in an engineered system. Understanding connectivity is critical to understanding system function and having the ability to detect and diagnose problems. Schematics provide simple, clear models of the systems they represent, allowing the problem

solver to envision connections and trace out and isolate components with faults.

For a large engineered system, there may be thousands of schematics. Rasmussen [1], in his book on interfaces for computer models of engineered systems, observed that coping with the complexity of these systems requires "a large repertoire of different mental representations of the environment." Davis [2], in an article on the use of intelligent systems on the Space Station, noted that good engineering depends in part on the judicious selection of the correct model for a given problem. In a real-time engineering context, not to mention the fragile environment of space, the user needs to have the best possible representation of the problem for maximum problem solving effectiveness.

Large numbers of schematics accumulate to depict an engineered system because engineers must be able to look at the system at different levels of detail, and from different perspectives. NASA engineers described to us four levels of schematic detail: the overview of the complete system, a view of each major subsystem, a view of a component area within a subsystem, and a detailed view of a single component. As one engineer put it, engineers need "information all the way down". Perspectives for a thermal system include mechanical, electrical and thermal perspectives. As problem solution proceeds, engineers move among the schematics to find the view that best captures that aspect of the problem they are working on at a given moment.

Because the Space Station is still in the design stages, we don't know how many schematics will describe it or its component subsystems such as the thermal bus. A comparable number for a nuclear power plant is about 2000 schematics in the "working set" (although the total number of

schematics for the plant may be much higher) [3]. For a particular problem then, finding the right schematic out of a large universe is an important aspect of solving the problem.

## INTERFACE PRECEPTS

One of the challenging things about building interface tools is figuring out what will ultimately help the end user of the interfaces developers build with the tools. Tools have a life of their own beyond the fact that they merely make it easier to construct something; more importantly, *tools define possibilities for what to build*. If developers have a tool for constructing cascading menus for example, suddenly we will see cascading menus appearing in our interfaces (for better or for worse). When tools suggest a good representation and provide useful functionality, they can be very powerful.

To maximize the problem solving effectiveness of our users as they work with large quantities of data, we are most concerned with providing tools that enable developers to build interfaces that help the end user to:

- Preserve focus
- Maintain orientation.

### Focus

As much as possible, users should be able to devote their attention to solving the problem at hand, not managing the computer. They should, in other words, be able to focus directly on getting their work done, with a minimal allocation of mental energy for interpreting information or taking action that does not bear directly on the problem solution.

The interface should be constructed so that it presents only the data of interest or relevance to the user for the phase of the task in progress at a given moment. The reduction of distracting

material has an important effect on the user's efficiency [4] and irrelevant information should be removed to reduce the effort of ignoring it [5].

### Orientation

In any system where users work with a large amount of data, it is easy to become disoriented, that is to lose a sense of where one is in the local or global environment [6], [7], [8]. In the world of the small, two-dimensional computer screen, the spatial cues used for orientation in everyday life no longer apply, or apply in extremely limited ways [9], [10], [11], [12]. The interface must introduce other conventions for orienting the user.

The orientation problem is of course a central concern for browsers, whose very purpose is to navigate around a large space. Users need to know where they are so that they can:

- Move to other places in the data space
- Interpret the meaning of unfamiliar information.

The possible meaning of say, an unfamiliar node on a graph, may be discerned by inspecting the nodes which are nearby and are likely to be similar, and by seeing where the node is in the total graph.

Also, to help users keep a sense of "place", an interface should present an aspect of stability in what is viewed. It is important to minimize the time the user spends adjusting to changing views, unless those views directly reflect some aspect of the problem data themselves.

### THE SCHEMATIC BROWSER

We deem a browsing tool an important part of an interface toolkit for our developers whose end users routinely deal with large quantities of data. It is not always possible for users to know

at the outset of a problem solving session exactly what they are looking for. Having the ability to browse quickly and easily should enable users to get to the schematics which best represent their problems.

Browsing is a visual activity. The user sees a set of choices, in some format, from which to select one to view in more detail. An unordered list exemplifies perhaps the simplest representation of choices. With this representation, however, lie two problems.

- There may be too many choices to look at.
- It may be too confusing to easily recognize a desirable choice.

One solution to the recognition problem takes advantage of structure inherent in the set of choices by relating them to one another in some way. Relations may be expressed, for example, in an ordered list, or a graph, as in Figure One.

The graph for the schematics in our application is a loosely-defined part-whole graph. Each node represents an actual schematic, and its children are schematics that show some part of the parent node's schematic in more detail. For example, the "Thermal Bus Overview" schematic (Figure Two) shows the three major parts of the thermal bus, viz., the evaporator (heat acquisition), transport and condenser (heat rejection) sections. The child nodes of the "Thermal Bus Overview" node on the graph (Figure 1) are "Evaporator Section", "Transport Section" and "Condenser Section". Each of the schematics for "Evaporator Section", "Transport Section" and "Condenser Section" shows its section in more detail than can be found in the "Thermal Bus Overview" schematic.

In looking at the graph, the user views nodes

ORIGINAL PAGE IS  
OF POOR QUALITY

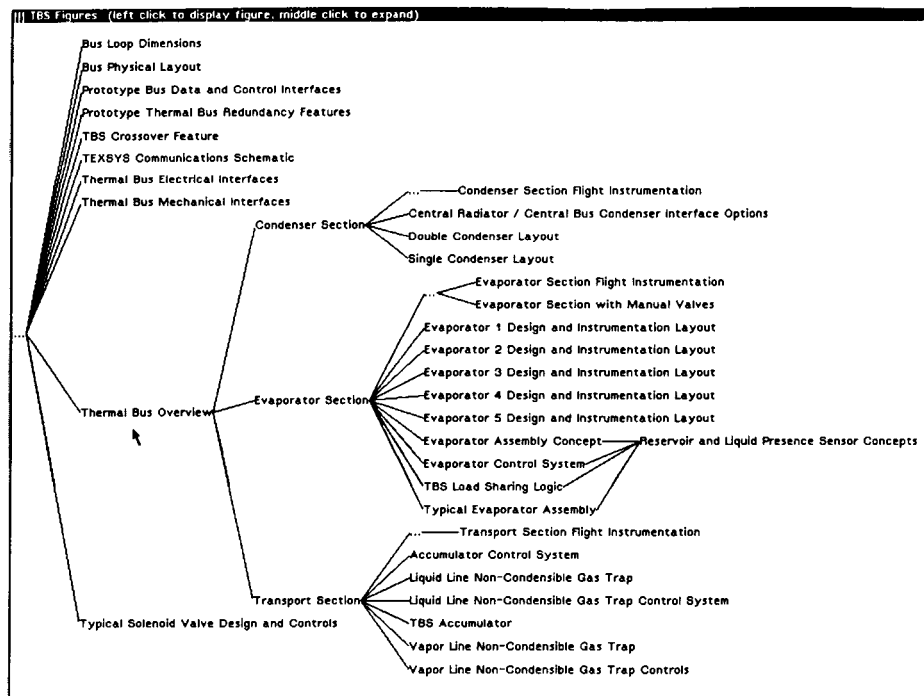


Figure One. Unfiltered part-whole graph of schematic diagrams. User selects a schematic for display by clicking on a node. The unfiltered graph shows too many choices for easy browsing.

ORIGINAL PAGE IS  
OF POOR QUALITY

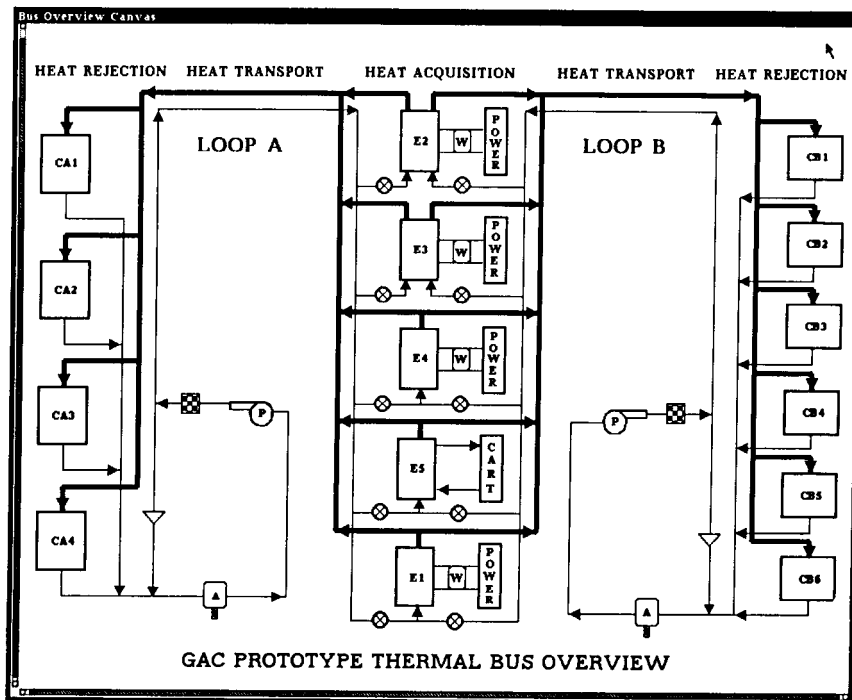


Figure Two. Thermal Bus Overview Schematic. A sample schematic being browsed. This schematic shows the major subsystems of the thermal bus: the condenser, transport and evaporator sections. Each subsystem has its own schematic in which it is shown in greater detail. In the graph, the subsystem schematics are children of the "Thermal Bus Overview" schematic.

which appear in the context of related nodes. The user finds significant orientation in the relations among the nodes so that interesting or relevant nodes are more apparent.

But too many choices remain. The graph lacks sufficient focus. Figure One for example has 36 choices, possibly too many to browse quickly and easily, and the number of choices in an application may of course be much greater. How can the interface filter the choices in an intelligent way and present the choices to their best advantage?

There is a natural tension between the need to filter to promote focus, and the need to provide context for orientation. To study this problem, much of our attention has turned to filtering the graph to a more manageable size, while at the same time providing sufficient context for the interpretation of the nodes.

Our general approach is to:

1. Filter out as many nodes as as possible
2. Add back in selected nodes for context.

#### **The Query-Filtered Graph**

First, the user or the intelligent system formulates a query to filter the set of choices and its graph down to a much smaller -- and more relevant -- subset. The query can be thought of as mapping a matching-predicate over the objects in the total set and collecting those that satisfy the predicate. For example, in Figure Three, the user has asked for a graph of schematics that show Evaporator-1. All of the nodes for the schematics containing Evaporator-1, that is, those that match the query, are shown.

In addition, the user sees two nodes that do not actually show Evaporator-1: "Condenser Section" and "Transport Section". As major

subsystems of the thermal bus, the "Condenser Section" and "Transport Section" are included as *landmarks* in the graph, providing recognizable geography or shape to the graph. Landmarks are static nodes, appearing in every graph, to provide a sense of large-scale structure. The user therefore views a graph of the matches for the query without losing a sense of the overall shape of the graph.

Non-matching ancestor nodes are also added back into the graph, again to provide the large-scale, global structural context for the matches. Ancestors provide shape and major reference points in the graph. In Figure Three, we do not see non-matching ancestors as all visible nodes are either matches or landmarks.

#### **Browsing**

A left mouse click on a node in the graph selects and displays the schematic represented by the node. A middle mouse click pops up a small window with auxiliary information about the schematic. (The auxiliary information could be any information relevant to the schematic such as page numbers in hardcopy editions, help, engineers' annotations, etc.)

The browsing environment has a Browser Window (see Figure Four) which displays user options and an alphabetical text list of the names of the components in the schematic currently displayed.

The schematics themselves are active. The user can click on a component to accomplish a variety of tasks such as further browsing (described in the following section), inspecting the underlying object in the knowledge base, filtering out the display components of its type from the schematic, showing a history of attached sensors, etc.

The user may also reformulate a query, refining

ORIGINAL PAGE IS  
OF POOR QUALITY

ORIGINAL PAGE IS  
OF POOR QUALITY

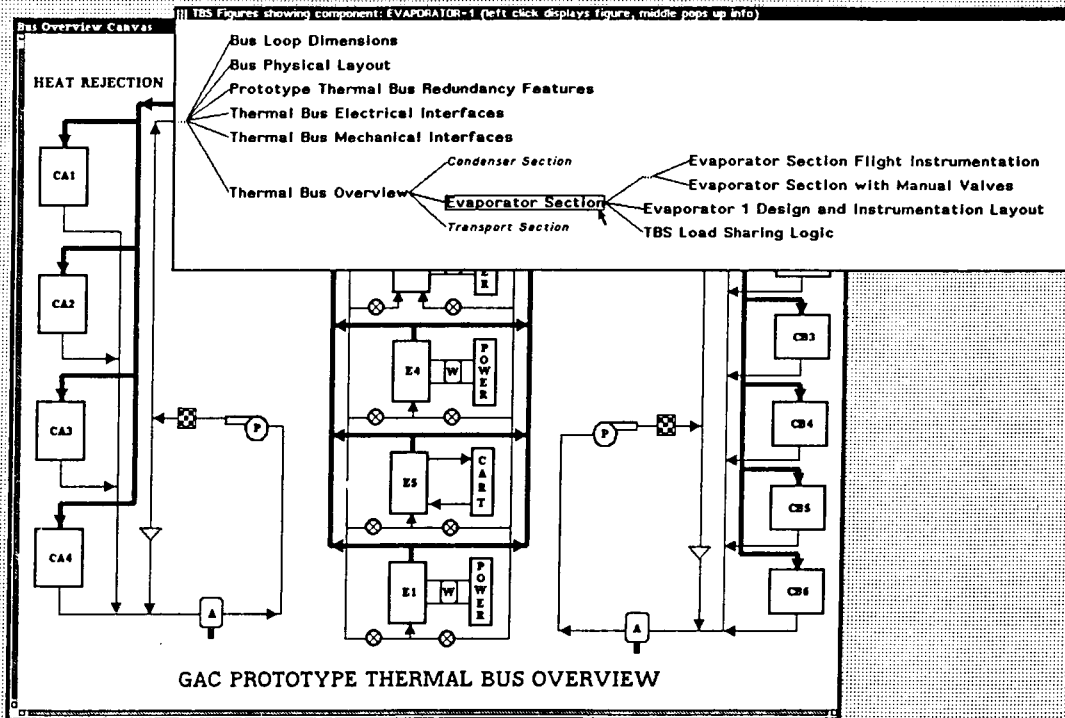


Figure Three. Filtered graph showing matches, in bold, for query requesting schematics with Evaporator-1. Landmarks, in italics, are a constant feature of the graph, giving a consistent shape to the graph for better user orientation.

it by asking for schematics with different or additional components. The user refers to components via a mouse selection on the picture of the component in the schematic, or by clicking on the name of the component in the text list. For example, the user may ask for only those schematics showing Evaporator-1, and refine the query to ask for schematics showing both Evaporator-1 and Evaporator-4. Many extensions for other kinds of queries can be envisioned, such as adding a perspective to the query, e.g., only those schematics of mechanical views showing Evaporator-1.

#### Adding Further Context

The query-filtered graph normally shows the user matches, landmarks and ancestors. However, the user may desire additional context to support the browsing activity. By its very nature browsing is interactive, and a user or intelligent system may not find the desired information the first time. At this point it may be helpful to show nearby nodes which are closely related to the actual matches. The Schematic Browser has an option for including *siblings* and *children* or just siblings of matches in the graph, as in Figure Five. The inclusion of sibling and child nodes develops a sense of *local context* for the user, so that nodes closely related to those that match the query are available for further exploration and to aid in the interpretation of the matches.

We also show a sort of dynamic local context by including the *previous matches* in reformulated queries, that is, the set of matches from the query immediately previous to the current one (Figure Six). For example, when the user reformulates the query for "Evaporator-1" as "Evaporator-1 and Evaporator-4", the previous matches from the query "Evaporator-1" are included. Thus the shape of the graph does not

change very much, helping to maintain consistency and orientation.

Other schemes for showing local context are of course possible, and may depend on application-specific criteria.

#### Presentation Control

In looking at the Figures in this paper the reader has certainly noticed the use of differing fonts to express node type on the graph. The fonts express *emphasis* and *de-emphasis* of the nodes, depending on their current status in the graph. Matches are emphasized - shown in large bold. The landmarks are smaller, italicized. The local context nodes are smaller still. The intention is to sharpen focus for the user by using: large fonts to draw the eye toward the nodes most likely to be of interest; smaller fonts to help reduce visual clutter; and the small italicized font to provide low-key emphasis for landmarks.

The user also has the ability to control the actual title used for the nodes. Optionally, shorter forms of the titles can be used for the de-emphasized nodes. In addition, page numbers (for the hardcopy versions of the schematics) are optionally shown.

To preserve the shape of the graph as much as possible, we maintain some standard order in the layout of the graph by sorting siblings alphabetically. This gives a feel of stability to the viewer so that nodes appear in the same context as new queries are formulated. (The alphabetical ordering is not optimal for the application and we will move to some new standard ordering to capture something of the semantics of the application, e.g., "Evaporator Section", "Transport Section" and "Condenser Section" in that order to reflect the functional flow of liquid in the bus.)

Again, many variations on the presentation

ORIGINAL PAGE IS  
OF POOR QUALITY



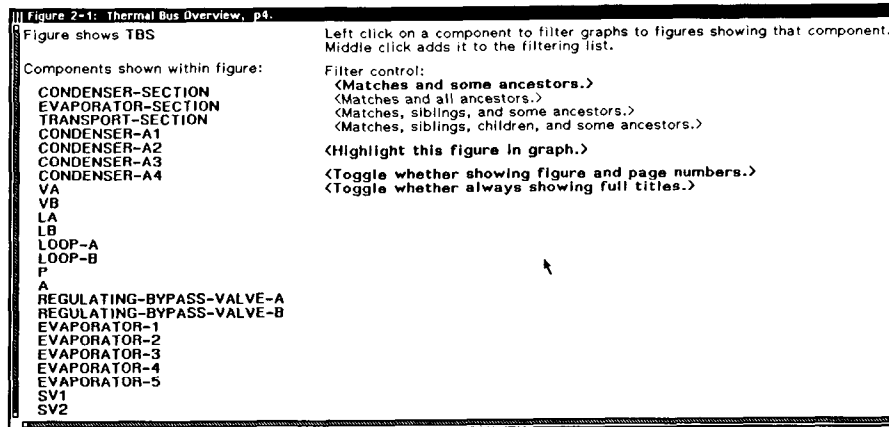


Figure Four. Browser Window for selection "Thermal Bus Overview". User options and a list of components in the schematic are displayed. The user may click on a component in the list for further querying.

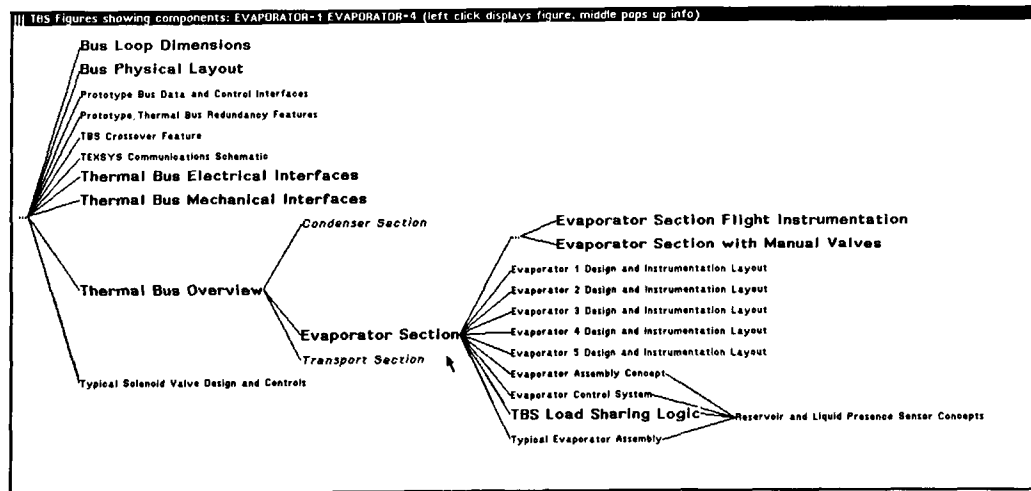


Figure Five. Filtered graph showing matches for query "Evaporator-1 and Evaporator-4", and siblings of matches, e.g., sibling node "Evaporator 1 Design and Instrumentation Layout".

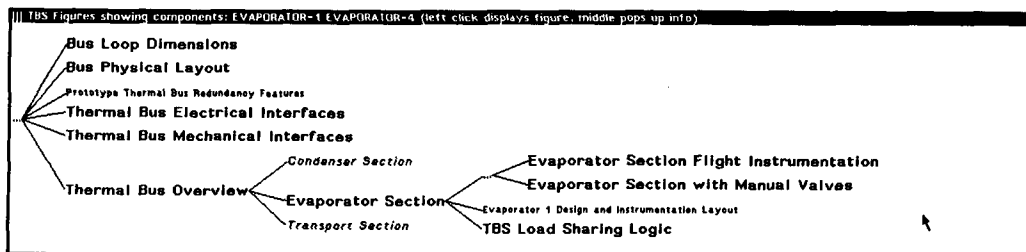


Figure Six. Filtered graph showing matches for query "Evaporator-1 and Evaporator-4", without siblings. Previous matches, nodes "Evaporator 1 Design and Instrumentation Layout" and "Prototype Thermal Bus Redundancy Features", de-emphasized in small font, are displayed from previous query for "Evaporator-1" (Figure Three).

control theme are possible. Experiments with the use of color or highlighting would prove interesting.

### **Fisheye Views**

Our work with filters and context management in graphical browsers falls into the category of generalized fisheye views described by Furnas [10]. The basic notion of a fisheye view involves a balance of local detail and global context. Furnas identifies the components of a fisheye view as a focus node and a degree of interest function composed of an *a priori* interest function and a measure of distance from the focus node. The degree of interest function is computed for the nodes of the graph, given the focus node, and the top *n* nodes are shown.

Our work fits this model, with some differences. First, we utilize a *set* of focus nodes -- the matching subset determined by the query. We do not have a notion of showing just the top-*n*-ranked nodes. We emphasize *a priori* interest rather than distance in selecting nodes for display, that is, landmarks, ancestors and previous matches. Only the optional inclusion of siblings and children one node away from matches is based on a distance measure. Our *a priori* interest function has a dynamic aspect not found in Furnas' work, in preserving the previous matches in the graph so that the current view is in part a function of its history. Our graphs tend more toward stability and consistent shape both in the long term because of the use of landmarks, and in the short term via the inclusion of previous matches. Feiner [8] notes that the drastic changes in the shape of a fisheye graph which commonly occur may be disorienting, so consistency aids offset a disadvantage of the fisheye approach.

Also, in our work, we have the addition of the use of presentation control for emphasizing and

de-emphasizing nodes. This seems to fit in well with the intuitive notion of a fisheye in which the eye is drawn to prominent parts of the view because of their size and clarity, while less important parts appear to be less distinct and more distant.

### **Integration of the Schematic Browser with an Intelligent System**

The Browser attempts to manage information presentation in a way that is supportive of the human user performing a browsing task. But significant portions of the browsing task may be done by an intelligent system which is monitoring activity in the domain knowledge bases and coordinating displays for the user.

As an example scenario of how this might work, the intelligent system and an astronaut are trouble-shooting a problem that appears to involve degraded condenser capacity. The intelligent system recognizes that the astronaut is likely to need a schematic view of the relevant subsystem. The intelligent system:

- Formulates an initial query, such as "Schematics showing condensers, their temperature sensors and the temperature sensors for the vapor lines".
- Chooses the options for the query-filtered graph -- whether to show children and siblings, full titles, etc.
- Chooses a schematic for initial viewing from the set of schematics which match the query.
- Annotates the schematic with messages and related displays, such as a temperature-time graph for the vapor line sensor whose reading is abnormally high.

The user can then proceed to use the selected schematic or refine the query and continue

**ORIGINAL PAGE IS  
OF POOR QUALITY**

browsing as needed.

#### SUMMARY

The complexity of today's engineered systems makes heavy demands on engineers and others who must access and interact with the massive quantities of information which describe the systems. Our paper has discussed some techniques for accessing the kind of schematic drawings used routinely in engineering problem solving.

Problems of information access include having too many choices from which to select something interesting and relevant, and losing track of where a choice fits into the overall organization of the information. In our browser, choices are presented in a graph to provide the user the orienting context inherent in the relations between the nodes. An important part of the work of the Browser is to present only those nodes likely to be useful to the user.

Our general approach is to filter the nodes down to a small relevant subset, and to then add back in selected nodes for context. The nodes in the graph are filtered by user-formulated queries, reducing the choices to a more manageable and relevant subset. The Browser establishes context for the nodes matching the query by adding landmarks, ancestor nodes, previous matches and optionally, siblings and children of the matches. The Browser's presentation control mechanisms help the user to focus on the most important nodes through emphasis of the matching nodes and de-emphasis of context nodes.

#### ACKNOWLEDGMENTS

We would like to thank the following people for their valuable feedback: Jane Malin of the NASA Johnson Space Center; and Conrad Bock, Brian Drummond, Victoria Gilbert, Catherine

Perman and Mike Williams of IntelliCorp. Errors are our own.

#### REFERENCES

1. Rasmussen, Jens, Information Processing and Human-Machine Interaction, Elsevier Publishers, New York, New York, 1987.
2. Davis, Randall, "Robustness and Transparency in Intelligent Systems", Proceedings Human Factors in Automated and Robotic Space Systems, National Research Council, Washington, D.C., 1987, 211-233.
3. Glazer, Keith, Southern California Edison, personal communication.
4. Kieras, David, "Guidelines for Diagrammatic Displays of Engineered Systems", unpublished manuscript, Ann Arbor, University of Michigan, 1987.
5. Bock, Conrad, "Visual Indexing and Interpretation or What's the Difference Between Frames, Logic, and Model Editors?", unpublished manuscript, Mountain View, CA, IntelliCorp, 1988.
6. IEEE Spectrum Staff, "Too Much, Too Soon: Information Overload", IEEE Spectrum, June 1987, 51-55.
7. Conklin, Jeff, "Hypertext: An Introduction and Survey", IEEE Computer 20:9, September 1987, 17-41.
8. Feiner, Steven, "Seeing the Forest for the Trees: Hierarchical Display of Hypertext Structure", Office Information Systems, Robert B. Allen, ed., ACM, 1988, 205-212.
9. Donelson, William, "Spatial Management of Information", Computer Graphics 12:3, August 1978, 203-209.
10. Furnas, George, "Generalized Fisheye

Views", Proceedings CIII '86 Human Factors in Computing Systems, Boston, April 13-17, 1986, 16-23.

11. Wise, John A., "Display Systems for Electrical System Control Centers", Proceedings of the Human Factors Society - 30th Annual Meeting, 1986, 1264-1268.
12. Shaw, John A., "Reducing Operator Error in Distributed Control Systems", InTech, March 1988.

ORIGINAL PAGE IS  
OF POOR QUALITY