Processes in Diagnostic Reasoning: Information Use in Causal Explanations

Dorrit Billman*+ (billman@psych.stanford.edu)
Daniel Shapiro* (dgs@stanford.edu)
Kirstin Cummings*+ (kirstinc@ccrma.stanford.edu)
Center for Study of Language and Information, Stanford University;* Institute for the Study of Learning and Expertise+

Abstract

In this paper we present examples of the processes people use in generating qualitative solutions to highly complex diagnostic problem solving. We developed a high fidelity model of the electrical power system for the International Space Station, and presented scenarios of off-nominal and fault situations. The model interface provides rich information about functional organization of the power system, including system topography and graphs of variables changing over time. We presented two versions, with system information organized hierarchically or displayed in a single level. Novices, who were unfamiliar with the system to be diagnosed but technically sophisticated, were asked to study the scenarios and diagnose the fault situations encountered. The particular scenario reported here was designed to be difficult, violate users' expectations, and require 'thinking outside the box.' Users chose to view quantitative information frequently as part of developing qualitative, causal explanations. We found sophisticated reasoning processes and frequently correct explanations despite the difficulty of the task. Design successes and weaknesses are discussed.

Introduction

We present examples of the processes people use in generating qualitative solutions to highly complex diagnostic problem solving. More precisely, we provide examples and analysis, but of a person-computer system. The computer presents a large amount of quantitative (variables) and topological (network) information. It does so in a way designed to display information selectively, and to help the person manage the complexity of information available. We ran a process-tracing study of problem solving and summarize here the variety of component procedures people used in the task.

This work fits in the tradition of complex problem solving research and the tradition of analyzing the affordances of human-computer systems. Our focus is on describing the procedures people used, how the procedures exploited the information available in the interface (particularly quantitative information), and the successes and pitfalls encountered. This paper reports on one problem scenario designed to be particularly challenging. In this scenario, the fault is outside of the system to be diagnosed. We expected that recognizing this might require breaking expectations about the form of the solutions. Troubleshooting a scenario that violates expectations is difficult, as when multiple fault scenarios require abandoning expectations about solution type (Patrick, Grainger, Gregov, Halliday, Handley, Fames, and O'Reilly, 1999). We thought that an expectation-violating scenario might provide a particularly useful window into the diagnostic reasoning supported by the system.

Domain and Tool

The power system of the Space Station is one of many complex systems that require ongoing monitoring and occasional troubleshooting. A high fidelity model of how the system behaves under a wide range of input conditions is a powerful tool for supporting these activities. In addition to high fidelity, a good model should be easy for people to understand and reason with. A transparent model (as opposed to a black box model) reveals the structure and relations among underlying components, which should make it easier to use. In particular, transparent models support diagnostic reasoning by less expert users. Experts often have internalized detailed models of a system, which let them reason from massive, unstructured information sets such as fluctuating arrays of variable values. In contrast, less experienced diagnosticians lack a detailed and fluent knowledge of how variables interact and affect each other. As a result, they cannot duplicate the expert's feat; they need information about the system from a source other than their background knowledge. Further, we believe that a tool that reveals the structure and function of the system being modeled would also aid experts. This has practical value as there is sometimes need for diagnoses to be done by less expert personnel. Economic needs to 'do more with less' and technical needs of extended duration missions will eventually require ground personnel or astronauts to monitor systems with which they are less familiar. Training technicians in structured troubleshooting methods, organized around functional subsystems, improves performance (Schaafstal, Schraagen, & van Berlo, 2000); we expected that our diagnostic system, which presents the structure and function of the underlying system, would support sophisticated diagnostic reasoning even by novices. Our system also provides information at multiple levels of scope and specificity, important in supporting troubleshooting for process control (Lindegaard, 1995).

We constructed the Power Monitor, a high fidelity, transparent model of the space station power system, and embedded it in a tool for monitoring and diagnosis. The modeling method we use represents the behavior of dynamic systems in terms of an interconnected network of processes and variables, called a *causal process model* [Langley et al., 2002]. Shapiro et al (2004) describes the Power Monitor in detail. Here we focus on its usability and the forms of diagnostic reasoning it supports.

Figure 1 shows the interface to the system. It depends on two primary representations of system information: a dynamic **network** of connected variables and processes, and **variable graphs** plotting values over time. Both provide much richer information than available in the current monitoring system and than typically provided in monitoring systems. The network nodes are the processes (rectangles) and variables (ovals); the links are arrows showing causal relations. Variables are linked to the processes for which they serve as input, and processes in turn are linked to their output variables. Thus the causal flow of the system is shown in a network of processes and variables. Processes and variables are flagged with a yellow (or red) border to mark deviation from predicted (or out of threshold) performance.

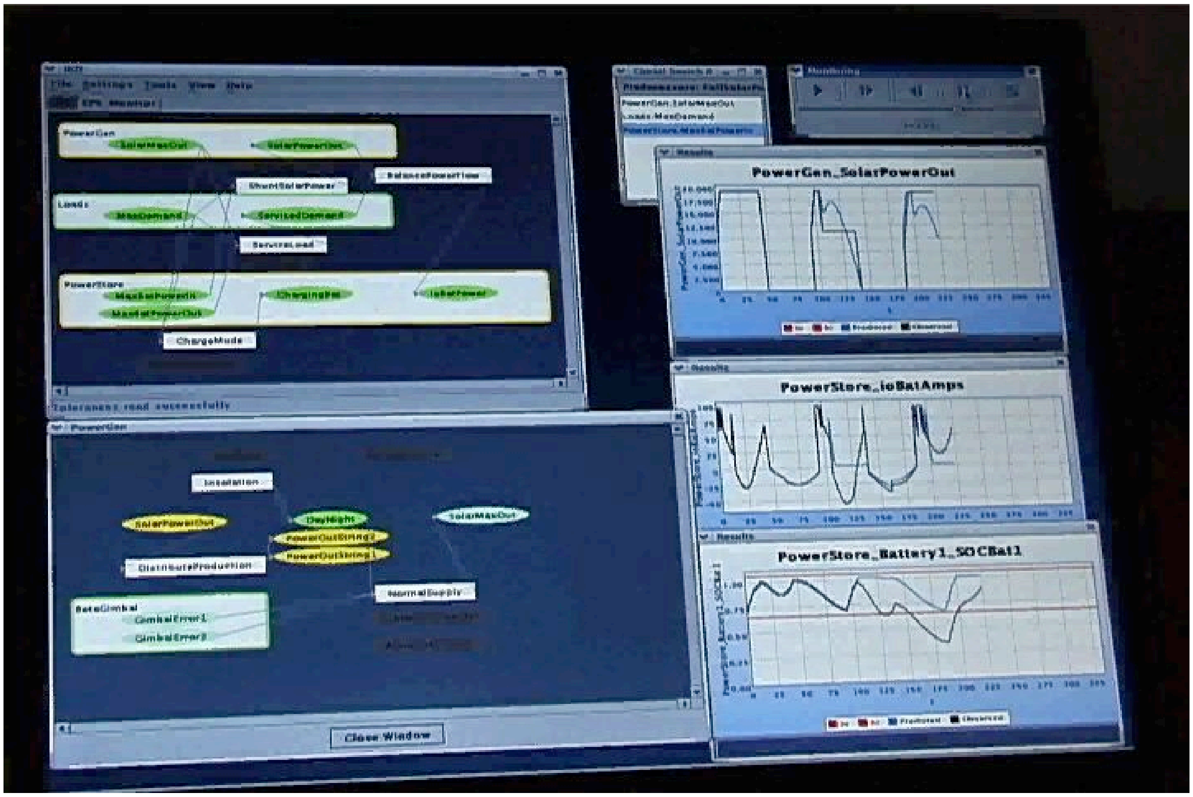

Figure 1: Hierarchical condition layout, with time set to Day 3, when generation is higher than predicted to compensate for under-generation on Day 2. User has 3 variables open and is comparing the timing and nature of the discrepancy between predicted and observed plots.

Two versions of the interface were used in the study, although comparison is not the focus here. In one version, the network was organized hierarchically. A top-level window showed subprocesses for the power generation, storage, and load subsystems. These could be clicked to show the process-variable network representing the subsystem, which in turn might have subsystems. In the flat version the network was displayed without any hierarchical grouping. In both cases the Power Monitor displayed the network by showing the flow of causal links from left to right, to the extent possible. In both cases the network required multiple screen-areas to show the entire layout. In the hierarchical condition, the user navigated through the display by clicking on subsystems and arranging the open windows. In the flat condition the user navigated through the display by scrolling across the whole layout to view the desired part of the network. In both, the network changes as different processes become active: only links to and from active processes are displayed and only active processes are highlighted. Thus, temporal navigation while viewing the network shows changes over time in the active processes.

Clicking on the variable oval opens the variable graphs. The graph displays the value of the variable (y) over time (x) from the beginning of the scenario, up to the current time step. Many variables are given both a directly observed value and a predicted value. The predicted value is what the variable would be if every thing were operating as planned. When the variable is as expected, the plot lines for the predicted and observed fall on top of each other. If the variable is not as expected, the observed values depart from those predicted. Using temporal navigation while viewing a variable graph will “draw” and “erase” the plot lines over time.

In addition to the video-like temporal navigation, the interface provides a method for “causal navigation”. Right-clicking on a variable or process node allows the user to show and then traverse either the set of forward links or backward links connected to that object.

Task and Participants

We used this testbed to look at complex, diagnostic problem solving by a human-machine system. Problems are scenarios in which off-nominal events -- serious or slight -- occur. A solution is a qualitative explanation of what is wrong, including identifying the root cause and the corresponding effects. Understanding effects is an important index of explanation quality, and also important because side effects can produce ancillary damage that needs to be addressed; for example, excessive discharge of the battery to compensate other faults can result in damage to the battery.

Even with very good diagnostic tools, locating and understanding faults in this domain can be very difficult. Even though the discrepancy between predicted and observed may be clearly flagged with a yellow border around a variable, it is a long way from noticing a collection of flagged variables to understanding the causal structure of the event. There are many system components and effects propagate over many links (creating breadth and depth); effects can be nonlinear because of compensatory interaction; faults can appear simultaneously at multiple components; and the time a fault is visible may be decoupled from the time the problem is flagged (because it may take multiple time steps to create the degree of discrepancy necessary to trigger flagging).

It addition to these complexity issues, problem solving is particularly difficult if it requires reasoning about situations beyond the presumed boundaries of the problem. People recognize in principle that information may be incomplete: sensors may fail and models can have errors. Nevertheless, it is very hard to simultaneously reason about an underlying system and “meta-reason” about one’s reasoning tools.

We hoped that novice users would be able to negotiate the diagnostic path if they were supported by the Power Monitor. We advertised in engineering classes at Stanford and on bulletin boards in the engineering buildings, for testers to use and evaluate the system. Our intent was to have users who were motivated, skilled in technical thinking, and familiar with at least some concepts relevant to system troubleshooting, electrical systems, circuit diagrams, and/or control systems. In short, we wanted people to diagnose a difficult, unfamiliar problem who were technically proficient but lacked knowledge about the particular system to be diagnosed.

The fault scenario we focus on in this paper is the Shadowed Panels Scenario. It was intended to require “thinking outside the box,” and was the first problem presented. The scenario simulated the situation in which the solar panels are partially shaded, as from an external object (or a piece of the Space Station) which begins to shadow the panels during the daylight (insolation) period, and stops during the night (eclipse). Thus the fault was actually outside of the target system. We thought this explanation would be hard to discover because the training that users had just received and the characterization of the experiment treated the Space Station power system as the target system to be diagnosed.

Study Overview: Methods and Results

Method. Twelve users participated; six tested the interface version with the hierarchical network layout and six tested the version with the flat network layout. The whole experiment lasted three hours. Participants worked on six problems, plus some auxiliary tasks. Users received training lasting roughly 40-60 minutes. We taught users about the general structure and function of the Space Station power system components, we explained and provided practice with the interface, and we gave some practice problems under normal operation conditions, such as identifying a good time to schedule an additional load and explaining why they chose that time. Users were asked to talk aloud during problem solving. Work times on the Shadowed Panels Scenario ranged from 7 to 32 minutes; users were urged to finish up after 25 minutes.

Results Summary. We summarize problem solving outcomes to provide context for discussing the processes used in this activity. Prior to the experiment we had identified two simple satisficing strategies which might produce explanations that users would find adequate. *Temporal Precedence* is a strategy of looking for the earliest component to be flagged as faulty, judging that component the cause, and all other flagged components as effects. *Causal Precedence* is a strategy of looking for the flagged component most upstream in the causal network, judging that the cause and all other flagged components as effects. Remarkably, no user restricted themselves to either of our simple, hypothesized strategies; all produced deeper and more elaborated accounts, and used more information. In all but one case, the user of the Power Monitor system produced a relevant diagnosis; 11/12 correctly localized the problem to the power generation functions of the system. Of greatest interest, in a third of the cases (4/12) the diagnosis was specific and correct: reduced sunlight. This required “thinking outside the box” in the sense that these explanations located the fault outside the focal system about which users were being taught and given data. The four users were able to compose the available information-gathering processes to produce a relevant, exact, expectation-violating diagnosis. A second measure gave users a list of possible characterizations and asked them to check the descriptions that applied; 10 of 12 checked “shadowing the panel.”

The data hints that the hierarchical interface supported diagnostic reasoning better than the flat interface. All the hierarchical condition users attributed the fault to the generation system; one to misalignment of the solar panels by the gimbal system (which rotates the solar panels to point at the sun) and five to reduced power generation; two of five focused on possible problems with shunting (deliberately reducing power generation) while the remaining three correctly concluded the panels were not getting enough sun due to shadowing by some object. In the flat condition, five users attributed the fault to generation, and one erroneously attributed the fault to unpredicted excess load. Of the five who identified the problem as generation, three focused on shunting, one on mechanical failure within the panels, and one on reduced input.

Explanations varied considerably in depth of understanding. Careful study of the protocols revealed one subproblem that gave us a very sensitive index of the sophistication of the user's model. This subproblem concerns the effect that appears on Day 3 as a result of reduced power generation on Day 2. Because of low generation on Day 2, the batteries were drawn down more than predicted. As a result, the power generation on Day 3 also is not normal: power is over-generated in order to recharge the batteries. Recognizing this over-generation and why it occurred requires a fairly elaborated and accurate model of the system dynamics in the scenario. The alternative user models of this subproblem included a) not noticing or analyzing this less critical departure from normal, or b) considering it a separate problem, e.g., caused by an independent episode in regulating shunting. Because all users' attention was focused on the more serious Day 2 problems, this is a difficult aspect of the overall problem.

Three of the six hierarchical users reached the correct and complete analysis of this sub-problem (two noted the over-generation but had different explanations; one did not note). In contrast, no user in the flat condition had the correct model, three never noted the discrepancy (either by cursor-pointing or by comment), two noted it but provided no explanation, and one provided an incomplete explanation. Developing the correct and complete model depended on a complex comparison. All users who discovered the correct solution compared the relation between predicted and observed values on one variable with the predicted-observed relation for one or more additional variables. Further, the solution required organizing the needed information: gathering operations and building an integrated model without becoming confused, disoriented, or overwhelmed.

What Processes Generated the Explanations?

Solving these diagnostic problems requires several types of process. The user must detect a fault, determine the scope of the problem in terms of the elements and time span involved, and understand the causal relations among these elements over this period. The user must navigate through an enormous amount of potential information in order to find the information that is relevant to the circumstance at hand. This requires understanding the information, integrating it to form an explanation, and modifying the explanation until either it seems satisfactory or further improvement seems unlikely. We focus our attention on the information gathering processes because these are the ones the interface was designed to support, and hence are the most observable. Our goal here is to sketch a taxonomy of the processes closely tied to gathering information,

We summarize here the basic operations supported by the interface for accessing information. We then focus on the more complex processes (composed of basic operations) that access and select information in the service of relational reasoning. Relational reasoning is a critical process because it is both closely linked to observable operations of information gathering, and is a key method by which information is organized to build a causal explanation.

From a complementary perspective, these processes show that users are capitalizing on the affordances of the Power Monitor to guide diagnosis. Participants use the graphs of variable values over time in sophisticated ways and in combination with network information. Users differ in how much they rely on variable information versus tracing status information through the network.

Basic operations. The system supports a set of operators for accessing visible information, network information, variable information, and the scenario as a whole.

1) Indicate and select information (standard GUI).

Actions: point with cursor to indicate any information and click or drag windows into position. Typical use: point to provide a visual anchor to any information being considered. Open and align windows to organize sets of information being used together.

2a) Navigate over the network: layout-based.

Hierarchical Action: open or close network subsystem window; arrange open windows. Flat Action: scroll network subsystem window to bring desired section of network into view. Typical use: to locate components marked as faulty by their color. Additional uses: to trace links in the network; to check what processes are active at a given time.

2b) Navigate over the network: causal links.

Action: right click to choose forward (effects) or backward (causes). Clicking on the tag for component X (variable or process) shows all components linked backward or forward; clicking on a tag highlights and displays the component in the network. Typical use: to find candidate effects or causes linked to a fault-flagged node.

3) View variable values.

Action: click on variable oval or move graph into view. Typical use: check the status of a variable. Often used in comparisons.

4) Navigate through scenario time. Actions: click to play; click to stop; click to move 1 time step; drag to target time-step. Typical use: play scenario through for initial viewing; play or drag over focal time of failure; step through critical period.

Composed Processes. The basic operations described above were composed into more complex, goal directed procedures. We identified six of the processes that people used to gather and reason with information. These are presented roughly in order of the complexity of information being used in reasoning.

1) Assess Network Status: View Fault-flagged Components. For many components, the model generated enough information to flag a component (by changing its color) if it was off-nominal. People used this information to

detect the occurrence of a problem, to give an impression of severity and change of severity over time, to bound the problem in terms of components involved, and to select variables for function-level viewing.

2) View Variable Function: Use Value-over-Time Representation of Variables. People used the displays of variable values plotted over time to reason in more detail about individual variables than supported by the network-level view.

a) Select variables for monitoring. Users checked the day/night variable to establish the overall pattern of activity for the power system. Similarly, they used the battery-charging graph to track the high-level power flow of the system. In the hierarchical condition, participants used the top-level window to select variables for monitoring, even though these variables were never flagged red or yellow. Interestingly, four of six hierarchical users opened unflagged, high-level variables from this window, apparently with a goal of monitoring or understanding the system rather than reacting to a particular problem variable

b) Diagnose from function shape. Users also studied the shape of the function to make very specific inferences. For example, one user used the step-function contour of the SolarPowerOut graph, at the point that arriving sunlight is cut and solar generation drops, to reason that the probable cause of the change was something outside the system:

"Here, at the beginning it goes as expected, and then suddenly, it drops. (pause) Things usually don't happen like this, like, it doesn't suddenly go into a right angle. It must be some kind of external thing."

3) Assess Discrepancy from Expected: Use Predicted Value Plotted with Actual Value. The availability of the plots of predicted values (and thresholds, when available), as well as actual values, supports a number of additional reasoning activities.

a) Scoping the problem. Users examined the paired plot lines to identify the time when one variable diverged from predicted value. They identified the point when 'things return to normal', using this to bound the time scope of a problem.

b) Type of Discrepancy. Users also determined the nature of the departure from a normal value, constraining the nature of the problem. At the end of Day 2, many users studied the discrepancy in SolarPowerOut to reason about the nature of the generation problem, with screens arranged as in Figure 1. For example, immediately after the point where the solar power drops, one user opened IOBatAmps (input/output Battery Amperes), and noted "here's a spike here [plays scenario] ... it's lower than expected." One particularly interesting case is the examination of SolarPowerOut when the generation on the third day is higher than expected, in order to compensate for the battery discharge on the second day. One user selected SolarPowerOut, looked at Day 3, started to say it was again too low, did a double take, and then corrected himself to say the power generation was now too high.

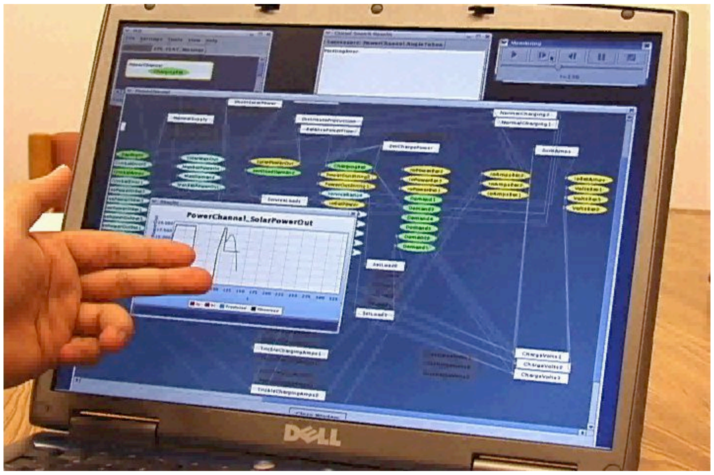

Figure 2. Reasoning about the discrepancy between predicted and observed on a single variable. This user rarely opened multiple variables at once, but worked through a series of off-nominal variables.

c) Hypothesis Rejection. Comparison of actual to predicted values also serves the very important function of allowing users to cleanly test and reject hypotheses. For example, once they had determined that generation was lower than it should be, several users hypothesized that the gimbal system might be responsible, and checked the gimbal variables. Finding that the actual values matched the predicted was a sufficient and compelling basis for rejecting the hypothesis that alignment of the panels by the gimbal was responsible for the problem. A few users also checked load variables to reject the possibility that excess demand was contributing to the problem.

4) Comparing Variables: Multiple Variables in View Simultaneously. Users opened multiple variable graphs at once, and compared them. Comparison was indicated both by talking aloud and by pointing to corresponding parts of two graphs.

a) Reference Comparisons. Many users related a reference variable to a second variable in order to develop a more integrated and coherent model of what was happening. Several users viewed the day/night graph to interpret what was happening in other graphs, such as SolarPowerOut. BatteryCharging was also used this way, as were the SOC (state of charge of the batteries) graphs.

b) Parallel Parts Comparisons. Users also compared the variables of analogous parts to see if a fault was local to one part or general to the system. For example, many users selected one variable, such as SOC (state of charge), for each of the three batteries. If the function looked the same for all three, users concluded that the problem was not specific to one battery, but originated outside and upstream of the individual batteries. Users then monitored variables from just one battery to track all three. Several users also did this in an analogous situation with two variables in the generation system.

5) Comparing Comparisons of Variables: Relating the Predicted to Observed Pattern in One Variable to Other

Variables. The variable representation supports a still more powerful type of reasoning, critical to understanding causal structure of the system. Users compared how and when one variable departs from predicted value with how a second variable departs from its predicted value in order to make complicated inferences about causal dynamics. To score behavior as "comparing comparisons" the user needed to relate predicted-observed information in one graph to predicted-observed information in the other, either by explicitly pointing between corresponding points on the two graphs, or relating the two variables verbally. An example screen layout is shown in Figure 2.

Six users clearly did this (five in hierarchical, one in flat); two additional (flat) users made comparisons between some variable and the binary day/night variable; two users (flat) never made multi-variable function comparisons and for two users activity was ambiguous but did not clearly show comparison. Users did these comparisons to determine which variable deviated from its predicted value first, and also to understand and reason about the compensatory relation between variables.

This user had opened SOCBattery1, SOCBattery2, and SolarPowerOut through the completion of Day 3.

"There was a deficit in solar power out [points to Day 2]. But here we have surplus [Points Day 3; pause] that could cure [points SOC Day 3] the problem of battery, to go back to its original predicted level."

6) Inferences from Process Information. Information about processes seemed to be harder to use than information about variables. Users did not always make the appropriate inferences about processes. Specifically, users might attribute a fault to a process even when that process was not flagged. For example, several users concluded that the fault lay in shunting because the process ShuntSolarPower was upstream of the problematic variable SolarPowerOut. This conclusion is suspect because the process was not fault-flagged. It would have been flagged if the expected input and output relations were not being maintained.

In contrast, noticing that this process was normal was the critical piece of evidence for one user to hypothesize that the problem must lie outside the system itself. This is one of the most sophisticated pieces of reasoning we observed, and critically exploits the information available about processes. [Here the screen layout was similar to that in Figure 1, but the cursor and attention were focused on the lower left window. The rectangular processes were all showing normal, but the "downstream" variables were yellow. User had checked the gimbal system, and concluded that it was fine.]

"They [the processes] are not lighting up either, uh, providing output for a given input. So, [sighs, pause] ad input equals bad output. Right input. [very long pause] All I can say is they're not getting enough sun. At this point."

Problems. Despite the successes reported here, the majority of users did not find and correctly integrate all the relevant information the system had to offer. Some users became lost or exhausted in the process. They might have known they did not understand everything but were uncertain how to proceed. As well as showing that people can make use of the resources offered in this interface, the study points to limitations of the design. The design provides an excellent model of the system being diagnosed, but it does not directly model or support the users' activity in solving the problem. For example, there is no support for keeping track of user-generated information: variables that have been examined, anomalies detected, hypotheses formed, or explanatory gaps remaining.

Conclusions

We were struck by the sophistication of the reasoning demonstrated by novices; this occurred in an area where human deficiencies are often conspicuous, especially in the absence of deep knowledge of the task. Although the study was conducted in the context of assessing one specific, real-world task, we think the demonstration of these reasoning processes is of broader consequence. They demonstrate successful reasoning with multi-variate, quantitative function information to develop causal explanations of problems in complex, unfamiliar systems. They illustrate the merits of designing tools for complex diagnosis that provide both rich topological and rich quantitative information. Sophisticated, successful problem solving emerges from the resulting human-machine system. Future analysis will identify more about the frequency and circumstances of using the various reasoning processes identified here.

Acknowledgments

The research reported in this paper was funded by Grant NCC 2-1220 from NASA Ames Research Center.

References

Langley, P., Sanchez, J., Todorovski, L., & Dzeroski, S. (2002). Inducing process models from continuous data. *Proceedings of the Nineteenth International Conference on Machine Learning* (pp. 347–354). Morgan Kaufmann.

Shapiro, D., Billman, D., Marker, M., & Langley, P. (2004). *A Human-Centered Approach to Monitoring Complex Dynamic Systems.* Final Report, NASA Grant NCC2-1220. Institute for the Study of Learning and Expertise, Palo Alto, CA.

Lindgaard & Gitte (1995). Human performance in fault diagnosis: Can expert systems help?_ *Interacting with Computers Special Issue: Australasian special issue: II United Kingdom Elsevier Science, 7(3),* 254-272.

Patrick, J., Granger, L., Gregov, A., Halliday, Handley, Fames, & O'Reilly (1999). Training to break the barriers of habit in reasoning about unusual faults. *Journal of Experimental Psychology, 5(3).*

Schaafstal, A., Schraagen, J., & van Berlo, M. (2000). Cognitive task analysis and innovation of training: The case of structured troubleshooting. *Human Factors, 42(1)*, 75-86.