Troubleshooting Strategies in a Complex, Dynamical Domain

Margaret M. Recker
T. Govindaraj
Georgia Institute of Technology
Atlanta, GA 30332-0280 U.S.A.
mimi.recker@cc.gatech.edu
tg@chmsr.isye.gatech.edu

Vijay Vasandani
International Business Machines Corporation
Marietta, GA 30067 U.S.A.
vijay@atlvm10.vnet.ibm.com

Abstract

In this paper, we present results from two empirical studies in which subjects diagnosed faults that occurred in a computer-based, dynamical simulation of an oil-fired marine power plant, called Turbinia. Our results were analyzed in the framework of *dual problem space search* (DPSS), in which non-routine diagnosis was characterized as a process of generating hypotheses to explain the observed faults, and testing these hypotheses by conducting experiments.

In the first study, we found that the less-efficient subjects conducted significantly more experiments, indicating a strong bottom-up bias in their diagnostic strategy. In the second study, we examined the effects of imposing external resource bounds on subjects' diagnostic strategies. Results indicated that constraints on diagnosis time led to a reduction in the number of actions performed and components viewed, without appearing to affect diagnostic performance. Constraints on the number of diagnostic tests reduced search in the experiment space, which appeared to negatively affect performance. Taken together, these suggest results that subjects' diagnostic strategies were sensitive to constraints in the external task environment. We close with a sketch of how DPSS might be augmented to include effects due to external resource bounds.

Introduction

The need for effective troubleshooting is rapidly becoming ubiquitous in our increasingly technological society. Troubleshooting is a complex cognitive process, requiring the integration of detailed system knowledge with strategies for locating, testing, and repairing faults under a dynamically changing environment. In this paper we attempt to understand and characterize this complexity. We present results from two empirical studies in which subjects diagnosed faults in a computer-based, dynamical simulation of an oil-fired marine power plant, called Turbinia [Vasandani and Govindaraj, 1993].

The data were analyzed in the context of a theoretical framework in which non-routine diagnosis is characterized as a process of generating hypotheses to explain the observed faults, and testing these hypotheses by conducting experiments. In cognitive science, such models have been described as *dual problem space search*, where processing alternates between search in the hypothesis problem space and search in the experiment problem space [Klahr and Dunbar, 1988].

In the first study, subjects' diagnoses were analyzed in order to determine their strategies for generating hypotheses, for conducting experiments, and for integrating search in the two problem spaces. In addition, we analyzed the efficiency of these strategies. Note that subjects were diagnosing faults with essentially no constraints on time and resources. However, in the real-world, resource bounds strongly affect the way diagnosis is conducted [Towne and Munro, 1988]. For example, a diagnosis situation may require that the failed device be repaired immediately, or a longer turnaround time may be permitted. Moreover, replacement parts may be plentiful and cheap, or they may be scarce and expensive. Finally, certain diagnostic tests may be difficult or time-consuming to perform. Therefore, in the second study, we imposed bounds on available resources in order to investigate their effect on subjects' diagnostic strategies. Two kinds of bounds were imposed: time and costs. The effects of time were investigated by manipulating the time available for diagnosing faults. The effects of cost were investigated by limiting the number of diagnostic tests. More theoretically, the goal of this study was to augment the framework to account for effects due to resource bounds during troubleshooting.

The remainder of the paper is organized as follows. In the next section, we present the theoretical framework underlying our studies. The following section describes the computer-based simulator, Turbinia. We then present the method and results.

Theoretical Framework

We define diagnosis as identifying the component that is causing the faulty condition. We propose that this process involves *identifying and clarifying* the initial symptoms, *generating hypotheses* to explain the symptoms, *running diagnostic tests*, and *evaluating* test results. Within cognitive science, the alternation between hypothesis generation and testing has been characterized as *dual problem space search* [Klahr and Dunbar, 1988]. In such models, search alternates between (1) the *hypothesis* space, which contains all possible hypotheses for the task, and (2) the *experiment* space, which contains all possible experiments for the task. Search in the hypothesis space entails proposing components whose failure best explains the observed symptoms. The search is guided by both prior knowledge and results from experiments. Search in the experiment space entails conducting diagnostic tests whose results, in turn, may confirm or disconfirm particular hypotheses under consideration. Search in the experiment space may be guided by currently active hypotheses, or may serve to gather information for formulating hypotheses.

In their studies of scientific discovery, Klahr and Dunbar (1988) have used this framework to characterize subjects in terms of how they search the two problem spaces. *Theorists* describe subjects who first attempt to generate hypotheses

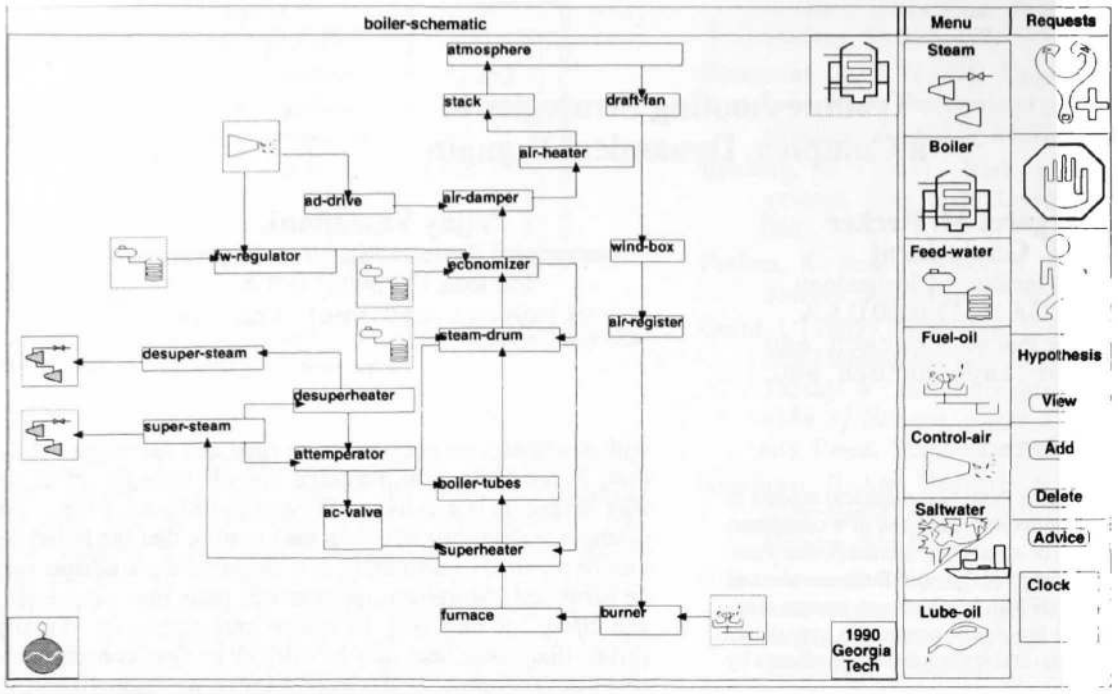

Figure 1: The boiler schematic in Turbinia.

that are then confirmed or disconfirmed through experiments. This top-down approach begins with search in the hypothesis space, and the search in the experiment space is constrained by the hypotheses under consideration. *Experimenters* describe subjects who search the experiment space without explicit hypotheses. In this bottom-up strategy, hypotheses are induced from experimental results.

Thus, strategies are crucial for determining where and how the two problem spaces are searched. For example, when searching the hypothesis space, are several hypotheses considered, or only one? When searching the experiment space, are tests conducted to 1) confirm the leading hypothesis, 2) disconfirm hypotheses, or 3) maximize information gain? In our empirical studies, we were interested in determining subjects' strategies for coordinating search in the two problem spaces, whether subjects could be characterized as theorists or experimenters, and the effects of resource bounds on subjects' search strategies.

Turbinia-Vyasa

Turbinia-Vyasa is an instructional system that trains operators in diagnostic problem solving in the domain of marine power plants. It is comprised of a steam power plant simulator and an intelligent tutoring system. Turbinia-Vyasa is implemented in Macintosh Common Lisp with Common Lisp Object System and runs on Apple Macintosh II computers. The simulator, Turbinia, is based on a hierarchical representation of subsystems, components, and primitives together with necessary physical and logical linkages among them. Turbinia can simulate a large number of failures in a marine power plant. Approximately 100 components have been modeled to achieve fairly high degrees of structural and dynamic fidelity even though the physical fidelity of the simulator is rather low. Vyasa is the computer-based tutor that teaches the troubleshooting task using Turbinia. The simulator, an interactive, direct manipulation interface, and the tutor (with its expert, student, and instructional modules) comprise the instructional system.

Vyasa operates in two modes: passive and active. In the passive mode the student is solely responsible for initiating the communications. When help is needed, the student must interrupt the simulation. She can then browse through and obtain detailed knowledge about the system and generic failures. When the passive tutor is invoked, the simulation is temporarily brought to a halt and the student can access various segments of knowledge in the expert module. In the active mode, the tutor takes the initiative to provide instructions when it infers a possible misconception based on the student's actions. In this mode, the tutor monitors the student's actions and offers help when appropriate. The instructions may be provided by the active tutor with or without intervention. The capabilities of active tutor include all the capabilities of the passive tutor as well.

A student interacts with Turbinia-Vyasa by choosing a schematic icon, a component, a gauge, or icons representing functions of the tutor. The boiler schematic, along with various icons, is shown in Figure 1.

Experiment 1

An empirical study was conducted with goal of determining the effectiveness of three training methods used by Turbinia-Vyasa. The details and results of this study can be found in Vasandani and Govindaraj (1993). In the analysis presented here, we focus on subjects' diagnostic strategies while diagnosing faults using only the simulator.

Mouse Category	Overall Mean	Mean Quick	Mean Slow	Main Effect $F(1,28)$	p-value	Group X Trials p-value
Gauges	25.19	16.76	33.42	6.85	.0001	.0001
Components	18.26	12.24	24.28	27.22	.0001	.02
Schematics	5.78	4.78	6.77	9.69	.005	n.s.
Symptoms	1.62	1.57	1.68		n.s.	n.s.
Diagnoses	4.00	3.57	4.42	-	n.s.	n.s.

Table 1: Mean number of mouse actions per group and per category, and ANOVA results.

Method

Subjects. Thirty Georgia Tech Naval ROTC cadets served as subjects, and were paid for their participation. All subjects had taken an introductory course on naval systems and had a basic understanding of thermodynamics.

Procedure. The experiment consisted of a training phase (10 sessions, each lasting approximately 1 hour) and a testing phase (two sessions). During the training phase, in which subjects diagnosed 28 faults, subjects were randomly assigned to one of three training conditions: (a) training using the simulator alone (Turbinia), (b) training with the aid of a passive tutor (passive Vyasa), and (c) training with the aid of an active tutor (active Vyasa). This was followed by an identical test phase (2 sessions), in which subjects diagnosed 10 faults, using only Turbinia. Five of these faults were new, while five had been given during the training sessions. The fault ordering was identical for all subjects.

Results

As previously mentioned, the present goal was to focus on subjects' diagnostic strategies. Since the differences between training conditions were small, we collapsed subjects across conditions, and only considered their performance during the final test (Turbinia only) phase, when they diagnosed the previously-unseen faults.

Turbinia kept a permanent record of each subject's mouse actions. Each mouse action served one of the following functions: 1) a request to view the initial fault symptoms (problem formulation or elaboration), 2) a request to view a schematic, 3) a request to view a component, 4) a request to view a gauge (hypothesis testing), and 5) a request to make a diagnosis (hypothesis evaluation).

The first mouse action was assumed to be problem formulation or elaboration. The second and third mouse actions were assumed to involve hypothesis formulation. Actions in the fourth category were assumed to involve hypothesis testing, while actions in the the fifth category were considered to be hypothesis evaluation.

Mouse Actions. Overall, the mean number of mouse actions per fault varied widely, and can be seen as reflecting fault difficulty. The high variability also suggested that there was no learning effect. For this reason, in the analysis, faults were sorted by difficulty (determined by the mean number of mouse actions per fault).

Table 1 shows the overall mean number of actions in each mouse action category. Viewing gauges was the most common activity, accounting for 39% of subjects' mouse actions. Viewing components accounted for 28% of subjects' mouse actions. Calls for viewing a new schematic accounted for 9% of the mouse actions, while evaluating diagnoses accounted for 6%. The most infrequent action was viewing the initial symptoms (2%). Thus, over a third of the subjects' actions involved conducting experiments.

Transitions. In Turbinia, the user can be in one of five diagnostic states: viewing a symptom, schematic, component, gauge, or requesting a diagnosis. At each state, the user can select from among the five possible actions (except the schematic state, in which only four kinds of actions are possible).

This results in a quite large state transition network, with 24 possible transitions. However, a much smaller portion of the network, with 9 possible transitions, accounted for the vast majority of subjects' state transitions. The reduced network did not show a transition between viewing the fault symptoms and requesting a diagnosis, suggesting that subjects did not rely upon symptom-fault associations. This is not surprising, since such pairs likely arise through gaining experience and developing sensitivity to common failures [Towne and Munro, 1988], and subjects in this study were lacking such experience. The most frequent transitions were *gauge-to-gauge* and *component-to-gauge* transitions (accounting for 39% of all transitions), again suggesting that subjects relied highly on testing.

The large number of tests suggested that subjects were following a strategy of attempting to confirm their hypothesis. The ubiquity of the positive-test strategy is a robust finding in the scientific discovery literature. While the positive-test strategy is generally acknowledged to be a less-efficient strategy than a negative-test strategy [Freedman, 1992], its optimality is, in reality, a function on the distribution of positive and negative instances [Klahr and Dunbar, 1988]. For example, if the probability of confirming a hypothesis is high, then a positive test result does not add much new information. However, in the Turbinia simulation, the majority of the gauges had normal levels. Therefore, subjects had a low probability of encountering abnormal gauges, which would serve to confirm their current hypothesis. As such, the use of a positive-test strategy is a quite reasonable heuristic.

Diagnostic Efficiency. Subjects were divided into two groups, *Quick* vs. *Slow*. The split was based on a post-hoc median split of the mean number of mouse actions performed when diagnosing novel faults during the test phase. In performing this split, we were interested in characterizing differences between efficient (Quick) and less-efficient (Slow) troubleshooters.

Analyses of variance were conducted with the number of mouse actions in each category per fault (sorted by difficulty) as the repeated measure, and group as the between-subjects factor. There was a strong main effect of trials (all p's $< .0001$) in *all* mouse action categories. This result again reflected fault

difficulty. On harder faults, subjects were simply making many more mouse actions in all categories.

In terms of the number of gauges viewed by subjects in the two groups (Quick or Slow), there was a strong main effect of group and an interaction of group with trials (see Table 1). Moreover, as faults increased in difficulty, the less-efficient group viewed significantly more gauges (there was a significant linear trend showing increased viewing of gauges with increased fault difficulty, $F(1, 28) = 24.66, p = .0001$).

Similar significant differences were found for the number of component viewings. There was a main effect of group and an interaction of group (Quick vs. Slow) with trials (see Table 1). Not only did the less-efficient subjects make significantly more component checks, their number increased significantly as faults became more difficult (there was a significant linear trend showing increased viewing of component with increased fault difficulty, $F(1, 28) = 13.07, p = .001$).

In terms of viewing schematics, there was a main effect of group, but no interaction of group with trials. There were no significant differences between groups nor interactions of group with trials in the number of times the initial symptoms were viewed or the number of diagnostic evaluations conducted (see Table 1).

Summary. By definition, the less-efficient subjects made more mouse clicking actions. An examination of their categories showed that the less-efficient subjects performed more mouse actions in all categories. However, a significant difference between less-efficient troubleshooters was found in the number of diagnostic tests performed. The less-efficient subjects conducted significantly more tests, and this difference became more pronounced as faults increased in difficulty. These subjects could thus be characterized as *experimenters* in that they were attempting to induce the failed component by searching for abnormal gauges. The more-efficient subjects conducted significantly fewer experiments, suggesting a better search of the hypothesis space.

Experiment 2

In the real world, many kinds of external constraints operate during diagnosis. In the second study, we investigated the effects of two kinds of external resource bounds on subjects' diagnostic strategies: time and cost. Time limits within Turbinia were implemented by restricting the time available to diagnose faults. Cost limits within Turbinia were implemented by adding a "Cost" window to the interface. Upon startup, this window displayed a number that was decremented by one each time a diagnostic test was conducted. In the case of Turbinia, this corresponded to consulting a gauge attached to a component.

Method

Subjects. Twenty-four Georgia Tech Naval ROTC cadets served as subjects, and were paid for their participation[1]. They were required to have the same background as subjects in the first study.

Procedure. The study consisted of three sessions, each lasting approximately two hours. The first session was a training phase, in which subjects diagnosed 8 faults. During the two test sessions, subjects were randomly assigned to one of four conditions: 1) time and cost bounds, 2) cost bound only, 3) time bound only, 4) no bounds. In each test session, subjects diagnosed 8 faults using only Turbinia. The fault ordering was identical for all subjects.

[1]The data from one subject were discarded due to an experimenter error.

			COST	
			yes	no
	Proportion	yes	.75	.56
	Diagnosed	no	.68	.90
	Time	yes	4.03	4.76
	(mins)	no	6.14	4.13
	Mouse	yes	39.37	44.79
	Actions	no	49.95	59.35
	Gauge	yes	9.95	14.56
TIME	Actions	no	14.00	17.85
	Component	yes	11.33	13.06
	Actions	no	16.18	18.05
	Schematic	yes	4.20	3.62
	Actions	no	5.72	4.40
	Symptom	yes	1.66	1.83
	Actions	no	2.18	1.70
	Diagnosis	yes	4.31	3.79
	Actions	no	3.64	5.65

Table 2: Proportion of faults diagnosed, time to solution, and means number of mouse actions per condition (yes=bounded; no=unbounded)

Design. There were two main factors of interest: diagnosis time (bounded, unbounded) and diagnosis cost (bounded, unbounded), resulting in a 2 X 2 between-subjects design.

Manipulated Variables: Time and Costs. Bounds on time and costs for each fault in the second study were determined by analyzing data from Study 1. From these data, we calculated the mean solution time and mean number of gauge actions per fault condition. For diagnosis time, subjects in the *unbounded* time condition were given 10 minutes to diagnose each fault. In the *bounded* time condition, for each fault condition, subjects were given the mean time to solution from the Experiment 1, rounded up to the nearest minute. For diagnosis costs, subjects in the *unbounded* cost condition were given 100 cost units for diagnosing each fault, an ample amount. In the *bounded* time condition, for each fault condition, the cost window was initialized with the mean number of gauge consultations from the previous study, rounded up to the nearest integer.

Results

In our analyses, we were interested in two performance measures (the number of faults diagnosed and the time to solution for faults successfully diagnosed), and several process measures (the total number of mouse actions and the number of mouse actions in the five mouse action categories). We conducted ANOVA on the performance and process measures, with COST and TIME as the independent factors. Means for each condition are shown in Table 2.

Performance. Despite imposing time bounds on some subjects, there were no main effects, but there was an interaction of COST with TIME on the time taken to diagnose faults, $F(1, 19) = 5.48, p = .03$.

There was a main effect of COST on the proportion of faults diagnosed, $F(1, 19) = 5.09, p < .05$, and an interaction

of COST with TIME, $F(1, 19) = 10.77, p < .005$. A Scheffé post-hoc analysis indicated that the subjects with cost bounds but no time bounds diagnosed significantly fewer faults than the subjects with both time and cost bounds ($p < .01$).

Number of Mouse Actions. Not surprisingly, there was a main effect of TIME, $F(1, 19) = 4.50, p < .05$. Restricting the time available to diagnosis faults appeared to reduce the resulting number of actions. However, restricting the number of gauge viewings (COST bounded) did not appear to affect the overall number of actions.

Viewing Gauges. Recall that the COST factor manipulated the number of gauge actions allowed. Predictably, there was a main effect of COST, $F(1, 19) = 3.75, p = .06$. However, there was not a main effect of TIME.

Viewing Components. There was a main effect of TIME $F(1, 19) = 3.94, p = .06$. However, there was not a main effect of COST.

Schematic, Symptom, and Diagnosis Actions. There were no main effects for these mouse action categories.

Summary. These results must be interpreted with care due to the small number of subjects in the study. However, the results suggested that subjects diagnosing faults with no resource bounds exhibited the most successful performance. This was not a surprising result. Interestingly, the subjects working under both time and cost constraints showed the next best performance. This suggested that the effect of working under extremely bounded conditions caused subjects to act more as theorists, rather than as experimenters. This conclusion seemed warranted given that these subjects made far fewer mouse clicks and consulted a much smaller number of gauges, whereas the differences between the number of actions in other categories were not significant.

Conclusion

In this paper, we have suggested that non-routine diagnosis can be characterized as search in dual problem spaces, alternating between hypothesis generation and testing. Search in the hypothesis generation problem space results in suspected components, while search in the hypothesis testing problem space serves to confirm or disconfirm specific hypotheses. Applying the DPSS framework, originally formulated to characterize scientific discovery, to troubleshooting tasks is important in suggesting common strategies underlying disparate problem solving situations.

Our analyses of subjects' diagnostic processes showed that they performed a large number of diagnostic tests. This suggests that subjects were primarily engaged in search of the experiment problem space. Unlike other studies where troubleshooters used strategies such as "half-split" or symptomatic search [White and Frederiksen, 1990], the subjects in the present study did not seem to engage in much symptom evaluation, and did not rely on symptom-fault pairs. This is perhaps due to their lack of experience in the domain, and the great ease of conducting tests in Turbinia.

The results of our analyses of subjects' diagnostic efficiency are consistent with those found in scientific discovery [Klahr and Dunbar, 1988]. We found that the primary difference between the diagnostic strategies of efficient and less-efficient subjects was in the number experiments conducted. The less-efficient subjects appeared to adopt a highly data-driven strategy of searching for abnormal gauges. This strategy could be viewed as a kind of abductive process, in which abnormal gauge readings allow the faulty component to be induced. The fact that the more efficient subjects performed significantly fewer diagnostic tests suggested that they engaged in a better search of the hypothesis problem space. It is possible that differences in prior knowledge about steam engines accounted for subjects' ability to formulate and effectively search the hypothesis space.

In the second study, we examined the effects of imposing resource bounds on subjects' diagnostic strategies. Subjects with no resource bounds exhibited the most succsessful diagnostic performance. Bounds on *time* allowed for diagnosing faults led to a reduction in the overall number of actions performed and components viewed, without appearing to affect performance. Bounds on the number of diagnostic tests (*costs*) reduced search in the experiment space, which appeared to negatively affect diagnostic performance. As suggested by this first study, testing was greatly relied upon by subjects (perhaps due to the structure of the Turbinia environment), and removing this capability adversely affected performance. Subjects with no time bounds but with cost bounds appeared to adopt a conservative diagnostic strategy, which in the end did not prove beneficial.

In our study, the cost factor (bounds on the number of experiments) appeared to have the largest effects on diagnostic efficiency and accuracy. We are currently working on augmenting the DPSS model to account for these results. This requires analyzing the role of experimentation within both DPSS and the task environment. In DPSS, experiments are conducted to generate or test hypotheses, or to gather data [Klahr and Dunbar, 1988]. In Turbinia, experiments consisted of reading gauges. Gauges had a fairly high density and, in the experimental conditions with no cost bounds, their access was cheap and easy. Our modeling approach involves adding an additional component to the search framework. Specifically, when considering a diagnostic test, the model must first estimate the cost of a particular diagnostic test against the expected information gain. As a model gains expertise, these estimates better reflect the cost structure of the task environment. We anticipate that this modeling approach will better capture the decisions and complexity faced by troubleshooters in real-world, resource-bounded situations.

Finally, these results also have design implications with respect to the kind of *fidelity* that is maintained between a computer simulation and its corresponding external system [Collins, in press]. Generally, designs attempt to maintain *epistemic* fidelity between the simulation and the external system. In the studies reported here, subjects' strategies were clearly sensitive to features and limits present in the diagnostic situation. As such, simulations of dynamical domains should attempt to preserve the costs and resource bounds of real problem solving situations.

Acknowledgements

This research is supported by the Army Research Institute for the Behavioral and Social Sciences under Contract No. MDA-903-90-K-112 to Janet Kolodner. The development of Turbinia-Vyasa was sponsored by the Office of Naval Research and contract N00014-87-K-0482 from Manpower, Per-

sonnel, and Training R & D Program. We thank Richard Catrambone and three anonymous reviewers for comments on an earlier draft of the paper. We also thank Michael Byrne for his advice on statistical matters.

References

[Collins, in press] Collins, A. (in press). Design issues for learning environments. In Vosniadou, S., deCorte, E., Glaser, R., and Mandl, H., editors, *International Perspectives on the Psychological Foundations of Technology-Based Learning Environments*. Springer Verlag, New York.

[Freedman, 1992] Freedman, E. (1992). Scientific inductions: Individual versus group processes and multiple hypotheses. In *Proceedings of the Annual Conference of the Cognitive Science Society*, Cambridge, MA. Erlbaum.

[Klahr and Dunbar, 1988] Klahr, D. and Dunbar, K. (1988). Dual space search during scientific reasoning. *Cognitive Science*, 12:1–48.

[Towne and Munro, 1988] Towne, D. and Munro, A. (1988). The intelligent maintenance training system. In Psotka, J., Massey, L., and Mutter, S., editors, *Intelligent Tutoring Systems: Lessons Learned*, pages 479–531. Lawrence Erlbaum Associates, Hillsdale, NJ.

[Vasandani and Govindaraj, 1993] Vasandani, V. and Govindaraj, T. (1993). Knowledge structures for a computer-based training aid for troubleshooting a complex system. In Towne, D., de Jong, T., and Spada, H., editors, *Simulation-based experiential learning*. NATO ASI Series F, Springer Verlag, Heidelberg.

[White and Frederiksen, 1990] White, B. and Frederiksen, J. (1990). Causal model progression as a foundation for intelligent tutoring systems. *Artificial Intelligence*, 42(1):99–157.