DOCUMENT RESUME

ED 192 743 IR 008 809

AUTHOR	Rouse, William B.; And Others
TITLE	Human Decision-Making in Computer-Aided Fault Diagnosis. Technical Report 434. Final Report.
INSTITUTION	Illincis Univ., Urbana. Coordinated Science Lab.
SPONS AGENCY	Army Research Inst. for the Behavioral and Social Sciences, Alexandria, Va.
PUB DATE	Jan 80
CONTRACT	DAHC19-78-G-0011
NOTE	29p.; For related document, see IR 008 801.
EDRS PRICE	MF01/PC02 Plus Postage.
DESCRIPTORS	*Computer Assisted Instruction; Computer Oriented Programs; Decision Making Skills; Difficulty Level; *Equipment Maintenance; Measures (Individuals); *Problem Solving; *Simulation; Task Analysis; *Training Methods
IDENTIFIERS	*Fault Diagnosis

ABSTRACT

A series of six experiments was conducted to increase understanding of human performance on diagnostic tasks, and in the process to investigate the feasibility of using context-free computer-based simulations to train trcubleshooting skills. Three simulated diagnostic tasks were developed: a simple context-free task, a complex context-free task, and a context-specific task (simulation cf aircraft pcwerplants). The six studies evaluated the effects of ccmputer aiding on the perfcrmance of each task and on subsequent unaided performance, using different task mixes, subjects (4 to 48 engineering or technical trainees), and conditions (self-pacing vs. forced pacing, feedback loops). Computer aiding reduced the number of tests required to diagnose simple problems and enhanced subsequent unaided performance except when subjects were under time pressures. Training on the simple task with computer aiding first inhibited and then enhanced performance on the complex context-free task. Training on the context-free tasks improved performance on the context-specific task. Results provide a database fcr bcth thecretical issues in fault diagnosis and practical application cf computer aiding to live system performance. (Author)

Technical Report 434

HUMAN DECISION-MAKING IN COMPUTER-AIDED FAULT DIAGNOSIS

William B. Rouse, Sandra H. Rouse, Ruston M. Hunt,
William B. Johnson, and Susan J. Pelligrino

Coordinated Science Laboratory, University of Illinois

Submitted by:
James D. Baker, Chief
MANPOWER & EDUCATIONAL SYSTEMS TECHNICAL AREA

Approved by:
Robert M. Sasmor, Director
BASIC RESEARCH

U.S. ARMY RESEARCH INSTITUTE FOR THE BEHAVIORAL AND SOCIAL SCIENCES
5001 Eisenhower Avenue, Alexandria, Virginia 22333

Office, Deputy Chief of Staff for Personnel
Department of the Army

January 1980

Army Project Number
2Q161102B74F

Basic Research in Decision Making

FOREWORD

The Manpower & Educational Systems Technical Area of the Army Research Institute for the Behavioral and Social Sciences (ARI) performs research and development in areas that include educational technology and training simulation with applicability to military training. Of special interest is research in the area of computer-based systems for maintenance training. The development and implementation of such systems is seen as a means of reducing time and costs by providing more highly individualized training than would be otherwise possible, while at the same time reducing the need for operational equipment for training.

This report summarizes a series of experiments conducted to increase our understanding of human performance on diagnostic tasks, and, in the process, to investigate the feasibility of using context-free computer-based simulations to train troubleshooting skills.

This research is responsive to the requirements of RDT&E Project 2Q161102B74F, "Basic Research in the Behavioral and Social Sciences."

Joseph Zeidner

JOSEPH ZEIDNER
Technical Director

HUMAN DECISION-MAKING IN COMPUTER-AIDED FAULT DIAGNOSIS

BRIEF

Requirement:

To investigate the effects of selected aspects of diagnostic tasks (problem complexity, pacing, and the presence or absence of computer aiding) on human performance. To investigate the effects of context-free diagnostic training on the performance of situation-specific diagnostic tasks.

Procedure:

Three diagnostic tasks were developed: a simple context-free task ("and" gates only); a complex context-free task ("and" gates, "or" gates, and feedback loops); and a context-specific task (simulation of aircraft powerplants). Six experiments were conducted to evaluate the effects of computer aiding on the performance of each task and the effects of aiding on subsequent unaided performance.

Findings:

Computer aiding reduced the number of tests required to diagnose the simple problems and enhanced subsequent unaided performance. The latter effect was not present when students were under time pressure, however. Training on the simple task, with computer aiding, first inhibited, then enhanced, performance on the complex context-free. Training on the context-free tasks improved performance on the context-specific task.

Utilization of Findings:

The results of these experiments provide a data base to be utilized for testing approaches to theoretical issues in fault diagnosis as well as the practical application of computer aiding to live system performance.

INTRODUCTION

This report summarizes research efforts aimed at increasing our understanding of human fault diagnosis abilities and how these abilities might be enhanced through the use of computer aiding. To this end, six experimental studies have been performed and three models of human behavior in fault diagnosis tasks developed. The results of this work are reviewed in this report. Also, future plans are discussed.

FAULT DIAGNOSIS TASKS

In choosing tasks around which experimental investigations could be based, several considerations were taken into account. First, tasks had to be reasonable, although perhaps somewhat abstract, representations of fault diagnosis situations that will be faced by real problem solvers. Second, tasks had to be representative of many different kinds of tasks. In other words, tasks specific to one particular piece of equipment were deemed undesirable. And finally, performance on the tasks had to be quantifiable such that comparisons among tasks could be more than a matter of opinion.

The three tasks that will be discussed here involve computer simulations of network representations of systems in which subjects are required to find faulty components. The three tasks represent a progression from a fairly abstract task that includes only one basic operation to another abstract task that includes two basic operations and, finally, to a fairly realistic task that includes several operations.

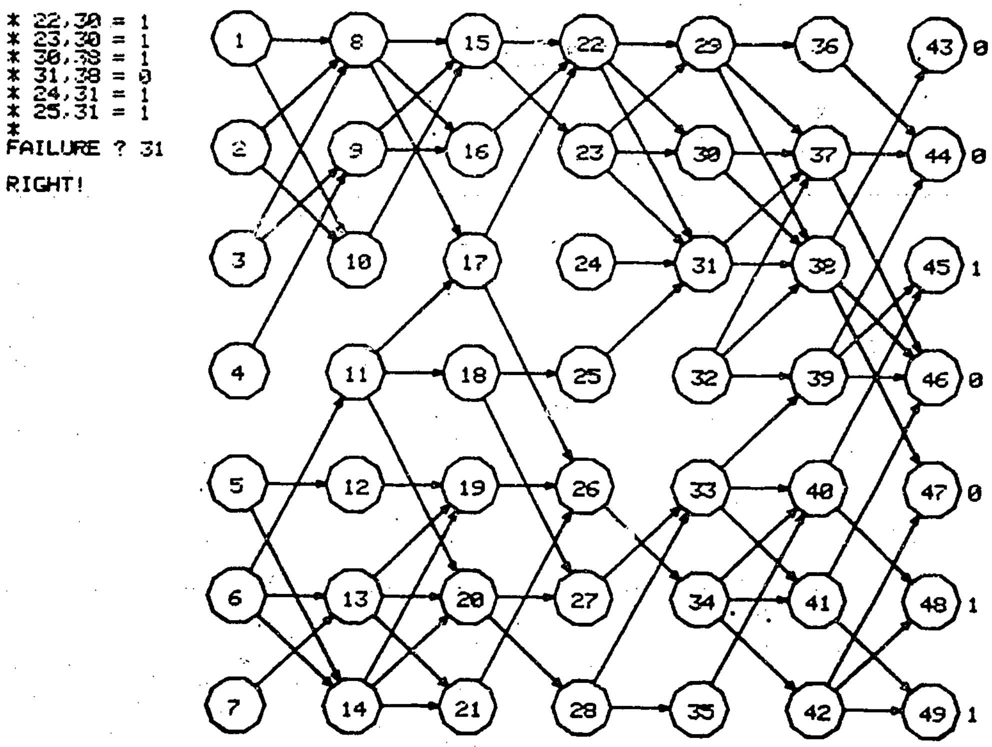
Task Number One

In considering alternative fault diagnosis tasks for initial studies, one particular task feature seemed to be especially important. This feature is best explained with an example. When trying to determine why component, assembly, or subsystem A is producing unacceptable outputs, one may note that acceptable performance of A requires that components B, C, and D be performing acceptably since component A depends upon them. Further, B may depend on E, F, G, and H while C may depend on F and G, etc. Fault diagnosis in situations such as this example involve dealing with a hierarchy of dependencies among components in terms of their abilities to produce acceptable outputs. Abstracting the acceptable/unacceptable dichotomy with a 1/0 representation allowed the class of tasks described in this paragraph to be the basis of the task chosen for initial investigations.

Specifically, the task chosen was fault diagnosis of graphically displayed networks. An example is shown in Figure 1. This display was generated on a Tektronix 4010 by a DEC System 10. These networks operate as follows. Each component has a random number of inputs. Similarly, a random number of outputs emanate from each component. Components are devices that produce either a 1 or 0. Outputs emanating from a component carry the value produced by that component. A component will produce a 1 if:

1. All inputs to the component carry values of 1,
2. The component has not failed.

If either of these two conditions are not satisfied, the component will produce a 0. Thus, components are like AND gates. If a component fails, it will produce values of 0 on all the outputs emanating from it. Any components that are reached by these outputs will in turn produce values of 0. This process continues and the effects of a failure are thereby propagated throughout the network.



```
* 22,30 = 1
* 23,30 = 1
* 30,38 = 1
* 31,38 = 0
* 24,31 = 1
* 25.31 = 1
*
FAILURE ? 31

RIGHT!
```

Figure 1. An Example of Task One

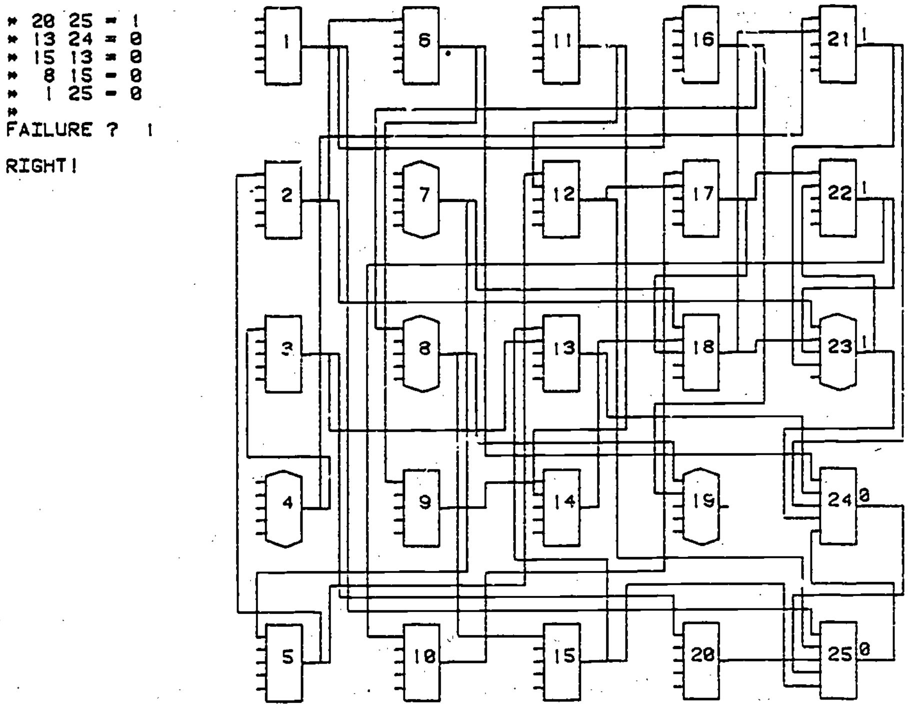
A problem begins with the display of a network with the outputs indicated, as shown on the righthand side of Figure 1. Based on this evidence, the subject's task is to "test" arcs until the failed node is found. The upper lefthand side of Figure 1 illustrates the manner in which connections are tested. A * is displayed to indicate that subjects can choose a connection to test. They enter commands of the form "component 1, component 2" and are then shown the value carried by the connection. If they responded to the * with a simple "return", they are asked to designate the failed component. Then, they are given feedback about the correctness of their choice. And then, the next problem is displayed.

In the experiments conducted using Task One, computer aiding was one of the experimental variables. The aiding algorithm is discussed in detail elsewhere (Rouse [11]). Succinctly, the computer aid was a somewhat sophisticated bookkeeper that used the structure of the network (i.e., its topology) and known outputs to eliminate components that could not possibly be the fault. Also, it iteratively used the results of tests (chosen by the human) to further eliminate components from future consideration by crossing them off. In this way, the "active" network iteratively became smaller and smaller.

Task Number Two

Task One is fairly limited in that only one type of component is considered. Further, all connections are feed-forward and thus, there are no feedback loops. To overcome these limitations, a second fault diagnosis task was devised.

Figure 2 illustrates the type of task of interest. Inputs and outputs of components can only have values of 1 and 0. A value of 1 represents an acceptable output while a value of 0 represents an unacceptable output. Thus, as with Task One, it is assumed that a situation with continuous inputs and outputs can be mapped into a representation such as that in Figure 2 using the acceptable/unacceptable dichotomy.



```
* 20 25 = 1
* 13 24 = 0
* 15 13 = 0
*  8 15 - 0
*  1 25 - 0
*
FAILURE ?  1

RIGHT!
```

Figure 2. An Example of Task Two

A square component will produce a 1 if:

1. All inputs to the component carry values of 1,
2. The component has not failed.

If either of these two conditions is not satisfied, the component will produce a 0. Thus, square components are like AND gates.

A hexagonal component will produce a 1 if:

1. Any input to the component carries a value of 1,
2. The component has not failed.

As before, if either of these two conditions is not satisfied, the component will produce a 0. Thus, hexagonal components are like OR gates.

The square and hexagonal components will henceforth be referred to as AND and OR components, respectively. However, it is important to emphasize that the ideas discussed here have import for other than just logic circuits. As a final comment on these components, the simple square and hexagonal shapes were chosen in order to allow rapid generation of the problems on a graphics display.

The overall problem is generated by randomly connecting components. Starting with component 1, and moving sequentially through the components, a random connection to another component is generated. Connections to components with higher numbers (i.e., feed-forward) are equally likely with a total probability of P_{FF}. Similarly, connections to components with lower numbers (i.e., feedback) are equally likely with a total probability of $P_{FB} = 1-P_{FF}$. The ratio P_{FF}/P_{FB}, which is an index of the level of feedback, was one of the independent variables in the experiments to be discussed later. In generating problems, two passes of all components are made. Thus, for example, up to 50 connections are possible with a 25 component problem. However, congestion in the layout sometimes causes the automatic connection router to fail and therefore, the maximum number of connections may not occur in a given problem.

OR components are randomly placed. The effect of the ratio of the number of OR to AND components was also an independent variable in the experiments to be discussed later. One interesting point to note is that an OR component with a single input is equivalent to an AND component with a single input. Since the random generation of connections does not assure that OR components will have multiple inputs, the effective OR/AND ratio varies even while the number of hexagonal components is fixed.

The task is performed by testing connections between components (see upper left of Fig. 2). Tests are of the form "component 1, component 2" where the connection of interest is an output of component 1 and an input of component 2. The subject's goal is to make tests until the faulty component is found. Further, since testing all components would be very time consuming, a procedure for choosing tests that will efficiently lead to the failure is desirable.

Task Number Three

Tasks One and Two are context-free fault diagnosis tasks in that they have no association with a particular system or piece of equipment. Further, subjects never see the same problem twice. Thus, they cannot develop skills particular to one problem. Therefore, we must conclude that any skills that subjects develop have to be general, context-free skills.

However, real-life tasks are not context-free. And thus, one would like to know if context-free skills are of any use in context-specific tasks. In considering this issue, one might first ask: Why not train the human for the task he is to perform? This approach is probably acceptable if the human will in fact only perform the task for which he is trained. However, with technology changing so rapidly, an individual is quite likely to encounter many different fault diagnosis situations during his career. If one adopts the context-specific approach to training, then the human has to be substantially retrained every time he changes situations.

An alternative approach is to train humans to have general skills which they can transfer to a variety of situations. Of course, they still will have to learn the particulars of each new situation, but they will not do this by rote. Instead, they will use the context-specific information to augment their general fault diagnosis abilities.

The question of interest, then, is whether or not one can train subjects to have general skills that are in fact transferrable to context-specific tasks. With the goal of answering this question in mind, a third fault diagnosis task was designed [Hunt, 1979].

Since this task is context-specific, we can employ hardcopy schematics rather than generating random networks online. A typical schematic is shown in Figure 3. The subject interacts with this system using the display shown in Figure 4. This alphanumeric CRT display was generated by a DEC System 10. The software is fairly general and particular systems of interest are completely specified by data files, rather than by changes in the software itself. Thus far, we have concentrated on various automobile and aircraft systems and, in particular, powerplant systems.

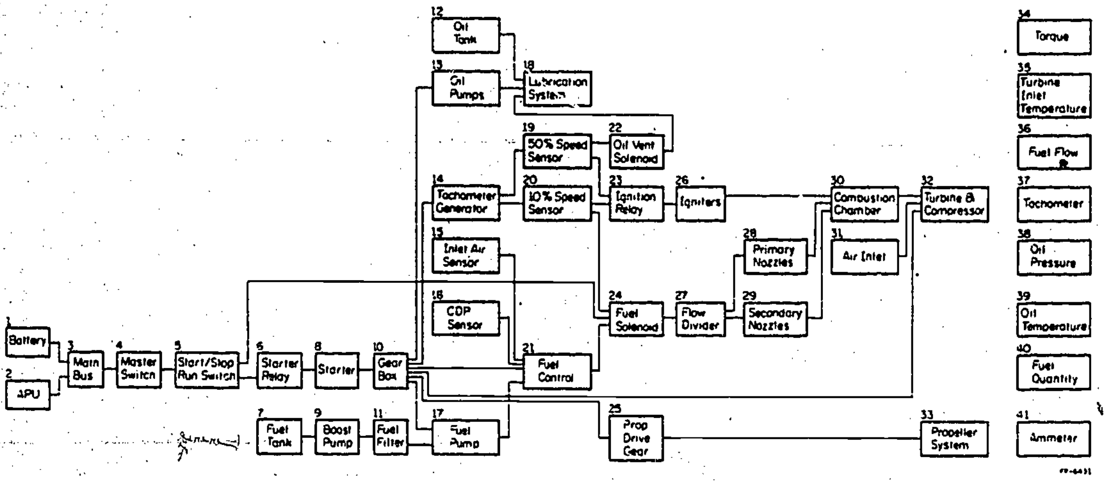
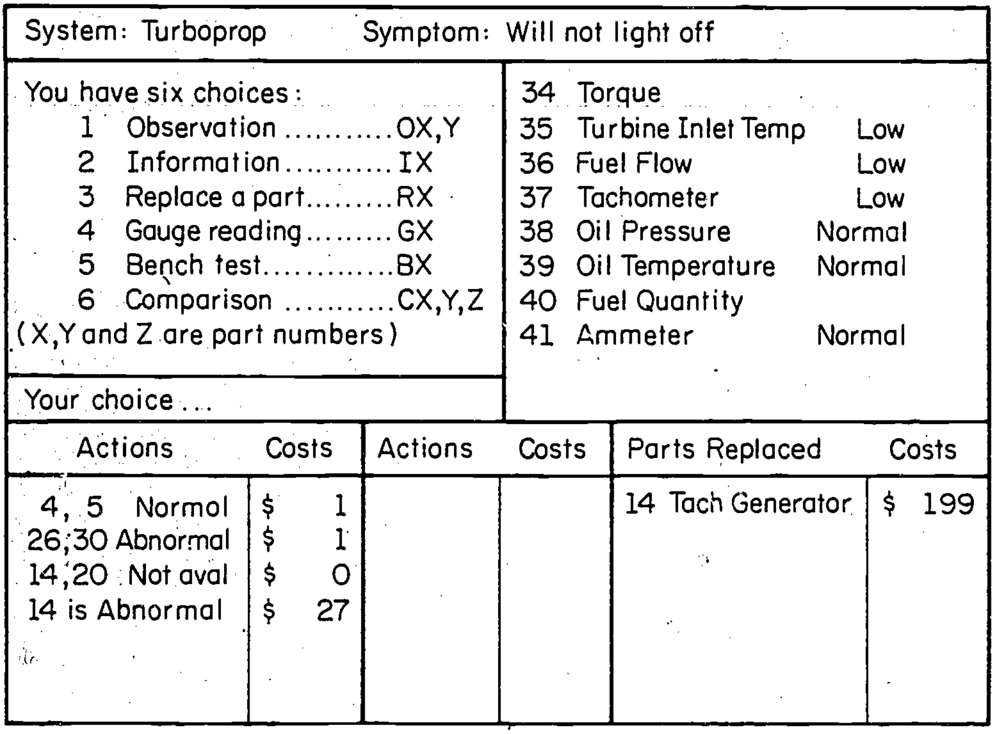


Figure 3. An Example of Task Three



System: Turboprop   Symptom: Will not light off

You have six choices:

1 Observation ....OX,Y
2 Information IX
3 Replace a part.........RX
4 Gauge reading.........GX
5 Bench test.............BX
6 Comparison ....CX,Y,Z

(X,Y and Z are part numbers)

Your choice...

34	Torque	
35	Turbine Inlet Temp	Low
36	Fuel Flow	Low
37	Tachometer	Low
38	Oil Pressure	Normal
39	Oil Temperature	Normal
40	Fuel Quantity	
41	Ammeter	Normal

Actions	Costs	Actions	Costs	Parts Replaced	Costs
4, 5 Normol	$ 1			14 Tach Generator	$ 199
26,30 Abnormal	$ 1				
14,20 Not aval	$ 0				
14 is Abnormal	$ 27				

Figure 4. Display for Task Three

Task Three operates as follows. At the start of each problem, subjects are given fairly general symptoms (e.g., engine runs rough). They can then gather information by checking gauges, asking for definitions of the functions of specific components, making observations (e.g., continuity checks), or by removing components from the system for bench tests. They also can replace components in an effort to make the system operational again.

Associated with each component are costs for observations, bench tests, and replacements as well as the a priori probability of failure. Subjects obtain this data by requesting information about specific components. The time to perform observations and tests are converted to dollars and combined with replacement costs to yield a single performance measure of cost. Subjects are instructed to find failures so as to minimize total cost.

Because the software developed for this task is very general, we feel that it will be used quite extensively for future investigations. In recognition of this flexibility, it seemed appropriate to devise an acronym. We concluded that an excellent acronym was FAULT which stands for Framework for Aiding the Understanding of Logical Troubleshooting.

EXPERIMENTS

Using the above tasks, six experiments have been completed, the first two of which were performed with support from a source other than the Army Research Institute. We will quite briefly review the results of these experiments.

Experiment One

The first experiment utilized Task One and considered the the effects of problem size, computer aiding, and training. Problem size was varied to include networks with 9, 25, and 49 components. The effect of computer aiding was considered both in terms of its direct effect on task performance and in terms of its effect as a training device [Rouse, 1978a].

Eight subjects participated in this experiment. Each subject solved six practice problems followed by three trials of 30 problems each. The experiment was self-paced. Subjects were instructed to find the fault in the minimum number of tests while also not using an excessive amount of time and avoiding all mistakes. A transfer of training design was used where one-half of the subjects were trained with computer aiding and then transitioned to the unaided task, while the other one-half of the subjects were trained without computer aiding and then transitioned to the aided task.

Results indicated that human performance, in terms of average number of tests until correct solution, deviated from optimality as problem size increased. However, subjects performed much better than a "brute force" strategy which simply traces back from an arbitrarily selected 0 output. This result can be interpreted as meaning that subjects used the topology of the network (i.e., structural knowledge) to a great extent as well as knowledge of network outputs (i.è., state knowledge).

Considering the effects of computer aiding, it was found that aiding always produced a lower average number of tests. However, this effect was not statistically significant. Computer aiding did produce a statistically significant effect in terms of a positive transfer of training from aided to unaided displays for percent correct. In other words, percent correct was greater with aided displays and subjects who transferred aided-to-unaided were able to maintain the level of performance achieved with aiding.

Experiment Two

This experiment utilized Task One and was designed to study the effects of forced-pacing [Rouse, 1978a]. Since many of the interesting results of the first experiment were most pronounced for large problems (i.e., those with 49 components), the second experiment considered only these large problems. Replacing problem size as an independent variable was time allowed per problem, which was varied to include values of 30, 60, and 90 seconds. The choice of these values was motivated by the results of the first experiment which indicated that it would be difficult to consistently solve problems in 30 seconds while it would be relatively easy to solve problems in 90 seconds.

This variable was integrated into the experimental scenario by adding a clock to the display. Subjects were allowed one revolution of the clock in which to solve the problem. The circumference of the clock was randomly chosen from the three values noted above. If subjects had not solved the problem by the end of the allowed time period, the problem disappeared and they were asked to designate the failed component.

As in the first experiment, computer aiding and training were also independent variables. Twelve subjects partcipated in this experiment. Their instructions were to solve the problems within the time constraints while avoiding all mistakes.

Results of this experiment indicated that the time allowed per problem and computer aiding had significant effects on human performance. A particularly interesting result was that forced-paced subjects utilized strategies requiring many more tests than necessary. It appears that one of the effects of forced-pacing was that subjects chose to employ less information in their solution strategies, as compared to self-paced subjects. Further, there was no positive (or negative) transfer of training for forced-paced subjects, indicating that subjects may have to be allowed to reflect on what computer aiding is doing for them if they are to gain transferrable skills. In other words, time pressure can prevent subjects from studying the task sufficiently to gain skills via computer aiding.

Experiment Three

Experiments One and Two utilized students or former students in engineering as subjects. To determine if the results obtained were specfic to that population, a third experiment investigated the fault diagnosis abilities of 40 trainees in an FAA certificate progam in power plant maintenance [Rouse, 1979a].

The design of this experiment was similar to that of the first experiment in that Task One was utilized and problem size, computer aiding, and training were the independent variables. However, only transfer in the aided-to-unaided direction was considered. Further, subjects' instructions differed somewhat in that they were told to find the failure in the least amount of time possible, while avoiding all mistakes and not making an excessive number of tests.

As in the first experiment, performance in terms of average number of tests until correct solution deviated from ontimality as problem size increased. Further, computer aiding significantly decreased this deviation. Considering transfer of training, it was found that aided subjects utilized fewer tests to solve problems and that they were able to transfer this skill to problems without computer aiding. A very specific explanation of this phenomenon will be offered in a later discussion.

Experiment Four

Experiment Four considered subjects' performance in Task Two [Rouse, 1979b]. Since the main purpose of this experiment was to investigate the suitability of a model of human decision making in fault diagnosis tasks that include feedback and redundancy, only four highly trained subjects were used.

The two independent variables included the level of feedback and the ratio of number of OR to AND components in a network of 25 components. Two levels of each variable were used in a within subjects factorial design. A latin square was used to determine the order of runs for each subject.

The results of this experiment indicated that increased redundancy (i.e.,more OR components) significantly decreased the average number of tests and average time until correct solution of fault diagnosis problems. While there were visible trends in performance as a function of the level of feedback, this effect was not significant. The reason for this lack of significance was quite clear. Two subjects developed a strategy that carefully considered feedback while the other two subjects developed a strategy that discounted the effects of feedback. Thus, the average across all subjects was insensitive to feedback levels. One of the models to be described later yields a fairly succinct explanation of this result.

Experiment Five

The purpose of this experiment was to investigate the performance of maintenance trainees in Task Two, while also trying to replicate the results of Experiment Three. Forty-eight trainees in the first semester of a two-year FAA certificate program served as subjects [Rouse, 1979d].

The design involved a concatenation of experiments Three and Four. Thus, the experiment included two sessions. The first session was primarily for training subjects to perform the simpler Task One. Further, the results of this first session, when compared with the result of experiment three, allowed a direct comparison between first and fourth semester trainees.

The second session involved a between subjects factorial design in which level of feedback and proportion of OR components were the independent variables. Further, training on Task One (i.e.,unaided or aided) was also an independent variable. Thus, the results of this experiment allowed us to assess transfer of training between two somewhat different tasks.

As in the previous experiments, Task One performance in terms of average number of tests until correct solution deviated from optimality as problem size increased and, the deviation was substantially reduced with computer aiding. However, unlike the results from Experiment Three, there was no positive (or negative) transfer of training from the aided displays. This result led to the conjecture that the first semester students perhaps differed from the fourth semester students in terms of intellectual maturity (i.e., the ability to ask why computer aiding was helping them rather than simply accepting the aid as a means of making the task easy).

On the other hand, Task Two provided some very interesting transfer of training results. In terms of average time until correct solution, subjects who received aiding during Task One training were initially significantly slower in performing Task Two. However, they eventually far surpassed those subjects who received unaided Task One training. This initial negative transfer and then positive transfer is an interesting phenomenon which we hope to pursue further.

Experiment Six

This experiment considered subjects' abilities to transfer skills developed in the context-free Tasks One and Two to the context-specific Task Three (i.e., FAULT). Thirty nine trainees in the last semester of a two-year FAA certificate program served as subjects [Hunt, 1979].

The design of this experiment was very similar to previous experiments except the transfer trials involved FAULT rather than the context-free tasks. Both Tasks One and Two were used for the training trials. Overall, subjects participated in six sessions of 90 minutes in length over a period of six weeks.

The results supported the hypothesis that context-free training can affect context-specific performance. For the two of the three powerplants used with FAULT, it was found that training with the computer-aided version of Task One reduced cost to solution, mainly because expensive bench tests were avoided and more cost-free information gathered.

MODELS OF HUMAN PROBLEM SOLVING PERFORMANCE

The numerous empirical results of the experimental studies discussed above are quite interesting and offer valuable insights into human fault diagnosis abilities. However, it would be quite useful if we could succinctly generalize the results in terms of a theory or model of human problem solving performance in fault diagnosis tasks. Such a model might eventually be of use for predicting human performance in fault diagnosis tasks and, perhaps for evaluating alternative aiding systems. More immediately, a model would be of use in focusing research results and defining future directions.

Fuzzy Set Models

One can look at the task of fault diagnosis as involving two phases. First, given the set of symptoms, one has to partition the problem into two sets: a feasible set (those components which could be causing the symptoms) and an infeasible set (those components which could not possibly be causing the symptoms). Second, once this partitioning has been performed, one has to choose a member of the feasible set for testing. When one obtains the test result, then the problem is repartitioned, with the feasible set hopefully becoming smaller. This process of partitioning and testing continues until the fault has been localized and the problem is therefore complete.

If one views such a description of fault disgnosis from a purely technical point of view, then it is quite straightforward. Components either can or cannot be feasible solutions and the test choice can be made using some variation of the half-split technique. However, from a behavioral point of view, the process is not so clear cut.

Humans have considerable difficulty in making simple yes/no decisions about the feasibility of each component. If asked whether or not two components, which are distant from each other, can possibly affect each other, a human might prefer to respond "probably not" or "perhaps" or "maybe".

This inability to make strict partitions when solving complex problems can be represented using the theory of fuzzy sets. Quite briefly, this theory allows one to define components as having membership grades between 0.0 and 1.0 in the various sets of interest. Then, one can employ logical operations such as intersection, union, and complement to perform the partitioning process. Membership functions can be used to assign membership grades as a function of some independent variable that relates components (e.g., "psychological distance"). Then, free parameters within the membership functions can be used to match the performance of the model and the human. The resulting parameters can then be used to develop behavioral interpretations of the results of various experimental manipulations.

Such a model has been developed and compared to the results of experiments One, Two, and Four [Rouse, 1978b,1979b]. The most important conclusions reached included:

1. The benefit of computer aiding lies in its ability to make full use of 1 outputs, which the human tends to greatly under-utilize,
2. The different strategies of subjects in experiment Four can be interpreted almost solely in terms of the ways in which they considered the importance of feedback loops.

It is useful to note here that these quite succinct conclusions, and others not discussed here [Rouse, 1978b,1979b], were made possible by having the model parameters to interpret. The empirical results did not in themselves allow such tight conclusions.

Rule-Based Models

While the fuzzy set model has proven useful, one wonders if an even simpler explanation of human problem solving performance would not be satisfactory. With this goal in mind, a second type of model has been developed [Pellegrino, 1979; Rouse, Rouse, and Pellegrino, 1979]. It is based on a fairly simple idea. Namely, it starts with the assumption that fault diagnosis involves the use of a set of rules-of-thumb (or heuristics) from which the human selects, using some type of priority structure.

Based on the results of Experiments Three, Five, and Six, we have found that an ordered set of twelve rules adequately describes Task One performance, in the sense of making tests similar to those of subjects 89% of the time. Using a somewhat looser set of four rules, the match increases to 94%. For Task Two, a set of five rules resulted in a 88% match. We have also found that the rank ordering of the rules is affected by training (i.e., unaided vs. aided).

The insights provided by this model led to the development of a new notion of computer aided training. Namely, subjects were given immediate feedback about the quality of the rules which the model inferred they were using. They received this feedback after each test they made. Evaluation of this idea within Experiment Six resulted in the conclusion that rule-based aiding was counterproductive because subjects tended to misinterpret the quality ratings their tests received. However, it appeared that ratings that indicated unnecessary or otherwise poor tests might be helpful.

Models of Task Complexity

It is interesting to consider why some fault diagnosis tasks take a long time to solve while others require much less time. This led us to investigate alternative measures of complexity of fault diagnosis tasks [Rouse and Rouse, 1979].

A study of the literature of complexity led to the development of four candidate measures which were evaluated using the data from Experiments Three and Five. It was found that two particular measures, one based on information theory and the other based on the number of relevant relationships within the problem, were reasonably good predictors (r=0.84) of human performance in terms of time to solve Tasks One and Two problems. The success of these measures appeared to be explained by the idea that they incorporate the human's understanding of the problem and specific solution strategy as well as the properties of the problem itself.

CONCLUSIONS

Within this paper, we have reviewed three fault diagnosis tasks, six experiments, and three models of human problem solving performance in fault diagnosis tasks. The empirical results indicate that humans have difficulty dealing with particular types of information (i.e., 1 outputs and, for some subjects, feedback loops). Further, the models have shown us how computer aiding can help subjects. Also, the empirical results have indicated that subjects can develop skills with computer aiding that are transferrable to situations where aiding is not available. Finally, we have found that context-free training can influence context-specific performance.

Beyond these results, the six experiments described here, when complete, will provide a data base for approximately 160 subjects and over 13,000 problem solutions. This data base should prove quite useful for testing initial approaches to various theoretical issues. For example, we plan to continue developing measures of complexity for fault diagnosis tasks. On a more applied level, our plans include a study of transfer of training from the three tasks discussed in this report to live system performance [Johnson, 1979]. As usual, all the research reviewed here has raised many more interesting questions, the answers to which are important if our knowledge of human problem solving performance in fault diagnosis tasks is to prove useful in the design of real-life systems.

REFERENCES

1. Hunt, R.M. "A Study of Transfer of Training from Context-Free to Context-Specific Fault Diagnosis Tasks," MSIE Thesis, University of Illinois at Urbana-Champaign, 1979.

2. Johnson, W.B. "Computer Simulations in Fault Diagnosis Training: An Empirical Study of Learning Transfer from Simulation to Live System Performance," PhD Thesis Proposal, University of Illinois at Urbana-Champaign, July 1979.

3. Pellegrino, S.J. "Modeling Test Sequences Chosen by Humans in Fault Diagnosis Tasks," MSIE Thesis, University of Illinois at Urbana-Champaign, 1979.

4. Rouse, W.B. "Human Problem Solving Performance in a Fault Diagnosis Task," IEEE Transactions on Systems, Man, and Cybernetics, SMC-8, No. 4, April 1978, pp. 258-271.

5. Rouse, W.B. "A Model of Human Decision Making in a Fault Diagnosis Task," IEEE Transactions on Systems, Man, and Cybernetics, SMC-8, No. 5, May 1978, pp. 357-361.

6. Rouse, W.B. "Problem Solving Performance of Maintenance Trainees in a Fault Diagnosis Task, Human Factors," Vol. 21, No. 2, April 1979, pp. 195-203.

7. Rouse, W.B. "A Model of Human Decision Making in Fault Diagnosis Tasks that Include Feedback and Redundancy," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-9, No. 4, April 1979, pp. 237-241.

8. Rouse, W.B. "Problem Solving Performance of First Semester Maintenance Trainees in Two Fault Diagnosis Tasks," Human Factors, Vol. 21, No. 5, October 1979.

9. Rouse, W.B. and Rouse, S.H. "Measures of Complexity of Fault Diagnosis Tasks," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-9, No. 11, November 1979.

10. Rouse, W.B., Rouse, S.H. and Pellegrino, S.J. "A Rule-Based Model of Human Problem Solving Performance in Fault Diagnosis Tasks," submitted for publication.