GRANT/ AMES

IN-54

62669

CR

P-86

NASA Grant NAG 2-123*

PILOT INTERACTION

WITH

AUTOMATED AIRBORNE DECISION MAKING SYSTEMS

Semiannual Progress Reports

March 1986 - August 1986

September 1986 - February 1987

John M. Hammer, Principal Investigator

Wan C. Yoon

Vijay Vasandani

Center for Man-Machine Systems Research

Georgia Institute of Technology

Atlanta, Georgia 30332

* The NASA Technical Officer for this grant is Everett Palmer, NASA-Ames Research Center, Moffett Field, CA 94035.

(NASA-CR-180258) PILOT INTERACTION WITH AUTOMATED AIRBORNE DECISION MAKING SYSTEMS Semiannual Progress Report, Mar. 1986 - Feb. 1987 (Georgia Inst. of Tech.) 86 p CSCL 05H

N87-19904

Unclas

G3/54 43504

INTRODUCTION

This report covers progress during two periods: March 1986 - August 1986 and September 1986 - February 1987. During this time substantial progress has been made in two areas. The first is Wan Yoon's Ph.D. thesis, "Aiding the Operator during Novel Fault Diagnosis." The second is a newer initiative, "A Model-based and Constraint-based Warning System."

The following were published during this period:

Journal articles

Yoon, W.C. and Hammer, J.M., "Aiding the operator during novel fault diagnosis," to appear in IEEE Transactions on Systems, Man and Cybernetics, 1987 (Appendix A).

Yoon, W.C. and Hammer, J.M., "A deep reasoning aid for aiding deep reasoning fault diagnosis," to appear in Human-Computer Interaction II, (G. Salvendy, ed.), Elsevier: Amsterdam (Appendix F).

Technical reports

Lewis, C.M., Identification of Rule-Based Models. Technical Report 86-5, Center for Man-Machine Systems Research, Georgia Institute of Technology, Atlanta, Georgia.

Conference papers

Yoon, W.C. and Hammer, J.M., "Aiding the operator during novel fault diagnosis," Proceedings of the IEEE 1986

International Conference on Systems, Man and Cybernetics, Atlanta, Georgia, 1986.

Technical Effort

During the first six months of this period, only Wan Yoon was supported. During the last six months, all three personnel were supported. During the summer of 1986, Dr. Hammer worked on the DARPA/AF Pilot's Associate program. Many of the interface concepts in the PA program were developed under NASA-Ames sponsorship. It was clear by the end of the summer that direct competition with this program was not possible. The PA program has more funding. The PA program can implement any aiding process that depends on knowledge acquisition from pilots. It cannot stop to answer basic research questions, although many have arisen during implementation. These unresolved questions are excellent topics for this grant because they are both relevant and realistic.

Relation to Earlier Work

The current research is focused on detection of human error and protection from its consequences. The first work in this area under this grant was [Hammer, J.M. "An intelligent flight-management aid for procedure execution," IEEE SMC 14(6), 1984], which described a program for monitoring pilot errors by comparing pilot actions to a script. There were two dimensions to this work. First, it dealt primarily with routine errors (slips) that occurred during checklist activity. Second, the model to which operator actions were compared was a script. There was no model of the aircraft or any part thereof.

Current research is an extension along these two dimensions. The ORS novel fault detection aid uses a sophisticated device model rather than a script. Since this aid has been used to study novel fault diagnosis, the errors committed are bad decisions, not slips. Although error detection is not currently implemented, the plans for it are discussed in [Yoon and Hammer, 1987] and later in this report.

The newer initiative, the model-based and constraint-based warning system, uses an even more sophisticated device model and is to prevent all types of error, not just slips or bad decisions.

PROJECT ORGANIZED BY MODELS OF DEVICES AND HUMANS

The principle that organizes this project is that model-based reasoning be the basis for aiding the human operator of an aerospace system. There are two models. First, the aid will contain a model of the device. The aid uses the device model to produce information for the operator. Second, the information produced for the operator is based on a model of human information processing. More specifically, the aid produces information that the operator needs and that is difficult to produce. What is difficult to produce is determined from the human information processing model.

The principle can be seen quite clearly in the novel fault diagnosis research. First, the aid has available to it a qualitative model of the orbital refueling system. Second, the aid uses this model to display information about what the ORS does normally (N aiding), what it is estimated to be doing (O aiding), and the difference between normal and observed behavior (O-N aiding). It can easily be argued that the unaided human operator must use at least some of this information in order to diagnose effectively a novel failure. This means the unaided operator must produce the information internally. It is difficult for the unaided operator to produce this information.

Model-based aiding can also be seen to organize the model-based and constraint-based warning system. The function of this aid is to keep track of the present and future constraints on the system and to detect present and future violations.

Central to this warning aid is a model of the physical system. Constraints arise from both physical and operational considerations. The model of the human is used to provide operational constraints and potential future inputs to the device. This model is actually part of the aid. The model also tells us that the operator does not or cannot consider all of the constraints when choosing an action. This model is not part of the aid. It is the reason that aiding was implemented.

Motivation for a Model-based Approach

The motivation for a model-based approach is two-fold. First, model-based aiding is aimed at a technological breakout through the use of artificial intelligence in device modeling and, to a lesser extent, in the operator intent inferencing. We believe this approach will yield larger system performance improvement than a more empirical approach.

The second motivation is to use what is known about human information processing and cognitive psychology to do cognitive engineering. Fortunately, exact predictions about human information processing are not always required. If some require processing is known to be difficult or error prone, then aiding (using artificial intelligence) should be investigated.

Artificial intelligence and cognitive psychology (or at least that part that is model-oriented) are close enough to use the same technical language. A consequence of this is a synthesis between the human and device models. Another consequence is an increased emphasis on the artificial intelligence technology of the aid.

AIDING THE OPERATOR DURING NOVEL FAULT DIAGNOSIS

The technical status of this effort is described in Appendices A and F. The remainder of this section describes the technical progress during the reporting period and future plans.

Technical progress

In February 1986, the ORS simulation was just a simulation connected to a display. There were plans for aiding, but no implementation. Since then, the following have been completed.

1. The code for O, N, O-N, and O-H aiding was written and debugged.

2. A preliminary, observation evaluation of unaided problem solving was conducted. The results are described in Appendix A.

3. Three experiments to evaluate N, O and O-N, and O-H have been planned. The first two have been completed and the results are described in Appendix F.

4. The training materials for the experiment were produced. The materials had to be carefully prepared and refined for two reasons. If they allowed too much practice or were otherwise too successful, the subjects might no longer use knowledge-based reasoning. On the other hand, too little training would not allow the subject to understand or interpret the basics of fluid flow.

Future plans for the ORS simulator

Considerable effort went into the construction of the ORS simulation. Relatively less effort was required to produce the existing aids. We would like to capitalize on this by studying a variety of research questions using the ORS simulator. The following are a list of potential problems to investigate.

1. Add and improve existing aids. We have observations from our more recent experiments that suggest more about aiding the operator. The O-N aid, which points out pressures which

differ between the observed and normal system, is useful primarily at the beginning of diagnosis. This is because it guides the diagnosis to the proper locale but is of less assistance thereafter. (In contrast, the O aid, which shows equal pressure paths and mass flow paths, appears useful throughout diagnosis.)

These observations suggest that operators need an aid during local testing. During local testing we have observed two operator deficiencies. First, the operator may incompletely test a local region (which does contain the fault), and then moves to another part of the ORS for testing. This greatly lengthens the time to diagnose. The operator needs to know when a locale has been completely tested. Second, the operator could probably benefit from seeing a list of suggested hypotheses. While suggesting hypotheses is computationally intractable for the entire ORS, it may be reasonable for small locales. In fact, if the aid were able to eliminate infeasible hypotheses as data were collected, the display of remaining hypotheses may keep the operator from leaving the locale prematurely.

Another observed problem is that the operator can choose good hypotheses but cannot effectively test them. This suggests a hypothesis testing aid which converts a specific hypothesis into a series of actions that test it. Interpretation of the results could optionally be included in the aid as well.

Another aid would prevent fault masking. It is possible to configure a malfunctioning system so that its fault is not apparent. For example, if there is a leaking valve, this failure can be masked by closing another valve in series with it. The operator can mask a fault through a series of changes and then be unable to unmask it. Unmasking requires only undoing all the changes, but the operator may not be able to remember them. It would be relatively simple to make available a command to return to the most recent state where abnormal behavior was observed.

2. Inference of operator intent, especially of hypotheses. For the aid to know the operator's intent would be useful in several advanced aiding methods described below. (Intent inference is not directly useful in and of itself.) The first step in understanding operator intent is to understand the process of hypothesis formation in the diagnosis task. While a preliminary description of this is in the paper in Appendix A, it is too subjective to be implemented on a computer. A more detailed examination of the verbal protocols currently being collected should yield a process description of diagnosis.

Given an objective process description, it would then be possible to detect the occurrence of decision-making biases during fault diagnosis. In fact, virtually all of the effort to do so is front-loaded into the intent inference work. Once an operator hypothesis is known, it would be relatively easy to test it for plausibility or keep track of how long the operator maintained it.

To build a training system for the ORS requires an intent inferencer. It may have to be modified to reflect a student's reasoning process. A more systematic approach, however, would be to let the intent inference be a prescriptive model or descriptive model of an expert. To accommodate the novice, a buggy model of dynamic process understanding could be used. This buggy model would be analogous to the buggy models of subtraction and programming that have already been developed.

3. A failure novel to the aid. Currently, the aid's model has a representation for every possible failure mode of every component. It would be interesting to give operators a failure that the aid's model does not

represent. This truly novel failure would occur after the operators had been aided on a series of more routine problems. It is important to know if the operators could determine when the aid was wrong.

MODEL-BASED AND CONSTRAINT-BASED WARNING SYSTEM (MCBWS)

MCBWS is a warning system for detecting present and future constraint violations in aerospace systems. For demonstration purposes, the fuel system of the F15 was chosen (Appendix E). The warning system contains a model of the fuel system and the physical and operational constraints on it. The purpose of this research is to demonstrate an electronic cocoon to surround the operator. The boundary of the cocoon is determined by present and future constraints. The system will be allowed to operate anywhere within the cocoon. Drawing near the boundary will cause an error message. Once demonstrated, the principles should be applicable to a wide variety of aerospace systems.

Motivation for the Warning System

Flying an aircraft requires thinking about the future. Avoiding error means avoiding constraint violations. Thus, it would seem that avoiding future constraint violations is central to avoiding error. Our view of flight is that it is a problem of remaining within the constraint envelope. The remainder of this section describes the implementation and current status of the project.

System Engineering

The two primary components of the warning system are the fuel system model and the constraint identifier. The fuel system model is capable of answering questions such as whether a particular constraint is currently violated or will be violated either now or in the near future. Predictions about a future constraint violation require both the constraint itself and likely fuel system inputs from now until the future point. Both of these come from the constraint identifier. Most constraints are the result of operator plans. The constraint identifier uses both the aircraft state and operator actions to select pilot plans. Associated with these plans are

- predictions of the future input actions to the fuel system
- constraints that must hold during the plan
- future plans that may occur, along with a description of the situation in which they will occur

As can be seen, once a plan is identified, its actions, constraints, and future plans are known. From future plans, future actions and constraints can be determined. Obviously, this forward chaining process can be continued as long as necessary or feasible.

Current Status

All of the technical effort has been devoted to building the fuel system model, which is more fully described below. The constraint identification code has received no attention because 1) I know how to do it from working on the Pilot's Associate program; 2) it is not hard to construct a plan recognizer for those plans relevant to the fuel system; and 3) the constraint recognizer cannot be tested without the fuel system model.

Fuel system model

The fuel system model is organized as a set of components. Each component is connected to other components or to inputs or outputs at the boundary of the system. Components have one or more behaviors, each of which is described by a set of equations or inequalities (termed algebraic relationships)[1]. These algebraic relationships describe the relationships between component inputs, outputs, and state variables. The relationships are symbolic and could be interpreted either quantitatively or qualitatively. If symbolic processing cannot answer a question about constraint violation, a numerical answer could be determined.

One of the basic operations is to solve the fuel system, which means to determine the behavior of each component. This occurs as follows. Each component has several mutually exclusive behaviors. First, find the subset of behaviors that is feasible. Some behaviors can be shown infeasible immediately because at least one algebraic relationship in the behavior is violated by other algebraic relationships known to be true. For each

[1] The constraints that arise from operational or physical considerations will be expressed in a form identical to the algebraic relationships that describe component behavior.

feasible behavior, assume that behavior is valid. Then, recursively attempt to solve the remaining components for their behaviors. This is a simple depth-first search through a space that is constrained by the behaviors of the components.

What has been written is the following. A slot-filler representation has been adopted for component descriptions (Appendix D). A set of routines that solves for the component behaviors has been written. A set of routines for manipulating a quantity space has been written. We were unable to reuse Wan Yoon's quantity space code for the following reason. His code uses property lists to store information. A change to a property, even if done within a function, is globally visible. It is as if properties are stored in global variables. Properties are not automatically undone during a search backup, which makes them undesirable.

Future Plans

The following must be done:

1. Build a model of the fuel system. This requires that we understand the fuel system: the types of pumps, the components that are not shown on the figure, etc. This understanding must then be encoded in the representation language and debugged.

2. Build the constraint identifier. This will require knowledge engineering with pilots to determine operational and physical constraints. These will be attached to plans, which will also require identification and duration conditions.

After this much development, the system can be demonstrated to detect current constraint violations. As described earlier, this is not sufficient to meet the need to prevent the consequences of pilot error. Reasoning about the future is also required. The second part of the project will develop this capability and will parallel the the first part.

1. Extensions for reasoning about the future. The model (the reasoning component, the device representation, and the fuel system description) must be enhanced to allow prediction of the future. The inputs to the model then become: the current system state, the predicted pilot inputs to the system, and the constraint(s) to be tested for potential future violation. It is possible for the model to output either yes or no. A no output means that there is no way that the constraint will be violated. A yes output will mean that there is no way to avoid violating the constraint. The most likely expected output from the model would be another list of constraints. This output list would have to hold for the input constraint to remain unbroken. The output constraints in general will have to hold at times not later than the input constraint. This is because if violating the output constraints would cause the input constraint to be violated, then the output constraints must be violated first. The output constraints hold at times closer to the present than the input constraints.

2. The constraint identifier knowledge representation will need to be enhanced. Each plan will also need to have (1)

potential future plans plus the conditions under which each future plan would occur; and (2) the predicted pilot input over the duration of the plan.

Appendices

A Aiding the Operator During Novel Fault Diagnosis

B Instructions for Wan Yoon's experiment, parts 1 and 2

C Problems worked after part 2 training (both training problems and experimental problems)

D Component knowledge representation

E F-15 fuel system

F A Deep-Reasoning Aid for Deep-Reasoning Fault Diagnosis

A

AIDING THE OPERATOR DURING NOVEL FAULT DIAGNOSIS

Wan C. Yoon and John M. Hammer

Center for Man-Machine Systems Research
Georgia Institute of Technology
Atlanta, Georgia 30332

ABSTRACT

The design and philosophy are presented for an intelligent aid for a human operator who must diagnose a novel fault in a physical system. A novel failure is defined as one that the operator has not experienced in either real system operation or training. Because the fault is novel, the human must reason using causal knowledge. The aid contains unique features that support such reasoning. One of these is a qualitative, component-level model of the physical system. Both the aid and the human are able to reason causally about the system in a cooperative search for a diagnosis. The aid has direct access to the operator's hypotheses when the qualitative model is used. Because of this, various decision-making suboptimalities and biases can be detected and mitigated by the aid.

INTRODUCTION

In highly automated systems, the human operator is primarily a monitor and supervisor [Rasmussen 1983, 1984]. An important monitoring function is diagnosing equipment faults, a difficult task in automated systems. The current approach to fault diagnosis is to train the operator to deal with relatively common faults. The training might teach the operator to use symptoms to distinguish faults and to follow procedures to correct them. While this approach should be successful with common faults, it does not support diagnosis of novel faults.

A common sense but unsuccessful approach to help operators diagnose novel fault is to teach them the principles of operation of the system. With this theoretical knowledge, the operators should be able, in principle, to diagnose any failure. Unfortunately,- there is little evidence that theoretical knowledge helps operators diagnose failures [Morris and Rouse 1985a, 1985b]. A logical consequence of this observation might be to put theoretical knowledge into the aid rather than the operator.

Recently, there has been much interest in supporting the human operator via expert systems for diagnosis. To be sure, this approach will improve the system performance on relatively common failures. As for novel failures, many expert systems for diagnosis [Shortliffe 1976, Miller, Pople, and Myers 1984] are based on shallow reasoning: a set of symptoms suggests a diagnosis. This mapping is not explicitly based on a system model. Consequently, such systems are subject to the same limitations as training and procedures. The designer may have to anticipate the failure for the expert system to solve it correctly.

Aiding from Deep Reasoning

In contrast to the above, our aid is based on deep, causal reasoning about the system. There are several advantages to this approach. First, novel fault diagnosis is normally considered to be knowledge-based reasoning [Rasmussen 1983]. Hence, it seems appropriate for an intelligent aid to reason causally. Second, this approach should be more reliable and robust. The system knowledge is represented at the component level. Because components are small and comprehendable, it should be possible to create representations that are correct, perhaps even provably so. These points support the belief that causal reasoning can cover a wider range of faults [Davis 1984].

In spite of the power of the intelligent aid, we believe there are several reasons to keep the human in command of the problem solving. First, diagnosing a novel failure may require the human to extend the aid's model. Second, when diagnosis involves operating the system (e.g., opening valves, starting motors), it would be better to leave these operations to the human. Third, causal reasoning is slow because the diagnosis problem is a combinatorial search. It may be that the human and the aid may be better able to find a solution cooperatively than either can alone. This is possible, even necessary, for two reasons. The human has better pattern recognition capabilities and can make inductive leaps. Second, the human may need to resolve ambiguities inherent in the aid's model.

Decision-Making Biases

The aid is designed to mitigate human suboptimalities that occur during decision-making and troubleshooting [Wickens 1984]. Two categories of suboptimalities used here are knowledge-limited and cognition-limited. The knowledge-limited suboptimality is simply that the operator does not fully understand the system. Obviously, the aid's model is a basis for compensating for this problem. There are many cognition-limited suboptimalities, which are discussed fully in a later section. The aid is designed, however, to prevent suboptimalities from occurring as well as detect and announce any that do occur. It should be noted that detection of suboptimalities requires a system model. Without a model it is logically impossible for the aid to interpret what the operator is doing. Thus, the system model is fundamental to aiding.

Motivation for this Research

There are several justifications and motivations for the research in this area. The first is to explore a new basis — qualitative models — for aiding humans in a domain for which there are few aids. Specifically, we wish to evaluate the suitability of qualitative models as the internal model of the aid. Many claims [Gentner and Stevens 1983; Rouse and Morris 1986] have been made that humans reason qualitatively about physical systems. The implication, which will be tested, is that qualitative models are useful as models in aids. Second, we wish to form a more detailed understanding of human diagnosis of novel faults. This presumably significant role for humans will be studied initially with observational methods, including verbal protocols.

In the subsequent sections of this article, we will review some relevant research on novel fault diagnosis, discuss the context of our experimental task, and discuss the qualitative model in our aid and its expected effects. In the final section, we will discuss the suboptimalities of interest and the methods to mitigate them.

REVIEW OF NOVEL FAULT DIAGNOSIS IN COMPLEX SYSTEMS

The literature on novel fault diagnosis in complex systems is limited. The section will have three parts. The first is empirical research on the effects of training on diagnosis. The second is Rasmussen's system engineering approach to the information needs of operators. The third is Wohl's performance model for predicting diagnostic times for novel failures. The last is the human information processing view of problem solving, which is similar in some ways to novel fault diagnosis.

Shepherd et al. [1977] have studied the effects of training on the errors operators committed while diagnosing familiar and unfamiliar failures. There were three kinds of training. The first was "no story," which amounted to a brief introduction to the control panel instruments. The second was "theory," in which the operation and flow of materials was explained. The third was "rules," which included the above theory training plus a set of proceduralized rules for diagnosing failures. After this training was administered, the three groups were tested. All three groups were significantly different, with rules best and the no story group worst on accuracy. The groups were then trained by examples to diagnose faults, and a second test revealed no differences between the groups. Later, all groups were tested again with two sets of faults — familiar and unfamiliar. Familiar faults were diagnosed equally well by all groups, but unfamiliar faults were diagnosed best by the rules group.

An experiment on the effects of training on operator control of a simulated process control plant has been conducted by Morris and Rouse [1985a]. One situation examined was the diagnosis of novel failures for which some of the subjects had sufficient theoretical training to diagnose the failure.

The system controlled was a network of fluid tanks. Fluid was pumped from these tanks through valves to neighboring tanks. Two novel failures were studied: a tank rupture that caused a loss in fluid, and a safety system failure that caused the system to shut down when it was not in danger. The experimental results did not show any differences due to training. Nearly all subjects were able to diagnose the tank rupture, and only half were able to diagnose the safety system failure.

System Engineering and Complex Diagnosis

Rasmussen [1983] has discussed operator control of complex systems in terms of three levels of information processing: skills, rules, and knowledge. Skill-based performance applies primarily to automatic, sensory-motor tasks that proceed without conscious control. One characteristic of such performance is that it is not decomposable or verbally expressible (for example, one cannot verbalize the skill of riding a bicycle).

The rule-based level is the second level of processing. A rule is a direct mapping from a set of input symptoms to a diagnosis or action. While performing at this level, the operator does not make recourse to causal models. Rule-based reasoning can be verbalized, which distinguishes it from the previous level.

The knowledge-based level is most relevant to the research reported here. Knowledge-based reasoning must be applied when novel failures occur. Neither skill-based or rule-based behavior should be used, and hopefully, the operator realizes this (but there is no guarantee). The operator's control occurs by first forming a goal and then a plan consisting of actions that lead to the goal. The plan is evaluated and perhaps modified by a combination of mental simulation or actual actions taken on the machine. Mental simulation relies, among other things, on the operator's mental model of the system.

Rasmussen [1985] has discussed functional and causal reasoning in diagnosis and control of complex plants during novel failures. Physical systems may be represented along a hierarchical, causal-functional continuum. The causal end of this dimension is a description of components according to their local behavior and their physical and structural location (much like a qualitative model). The functional end of the dimension is a description of aggregates according to their function or purpose. In highly automated systems, the operator also needs to know the intent of the automation, since it can change both the function and structure by its own action. The implications for novel fault diagnosis are the claims that an operator needs a multilevel display for intention, function, and causation. The motivation for this is that diagnosis begins at a functional level and moves toward a causal level as the diagnosis becomes more precise.

Maintenance Complexity

Wohl [1982] has observed that electronic troubleshooting in complex equipment operates in two modes. This first mode is for routine failures, which account for 65-80% of all failures. These are repaired relatively quickly. The second mode is for novel failures, which require substantially more time to diagnose and lengthen substantially the mean time to repair. A model for predicting the frequency distribution of novel malfunction repairs has been developed and tested. The model has three parameters: an equipment complexity index, which is the average connectivity of a component; second, an average time to test a component; and third, a parameter that describes how diagnostic interpretation becomes geometrically more complex with each diagnostic test. The test of the model showed a correlation of r=.98 between measured and predicted mean time to repair for fourteen different electronic systems. In a related article, Wohl [1983] observed that the model predicted an infinite mean time to repair when the equipment complexity index exceeded 7.5. An infinite mean time to repair simply means that some malfunctions are never diagnosed. An equipment complexity index of 7.5 means that the average component is connected to 7.5 other components. This limiting value is close to the chunk capacity of human working memory. This result is consistent with the often observed relationship between connectivity and diagnosis complexity.

Complex Diagnosis and Human Problem Solving

Much of the research on problem solving would appear to be relevant to novel fault diagnosis [Newell and Simon 1972]. We briefly review here the human information processing approach to modeling of problem solving and then discuss how novel fault diagnosis differs from it. The information processing approach is centered around the idea of a problem space, which is the human's representation of the key characteristics of a problem. The subject is given an initial and goal state in the problem space and a set of operators that transform the problem from one state to another in the problem space. Usually, the states and operators are crisply defined. Often, there is a metric for the difference between a given state and the goal state. This metric can be used as a heuristic for selecting the operator that moves the greatest distance toward the goal.

The behavior of a human is modeled by a production rule system. Each production rule contains a condition and an action. The condition is a boolean expression on the features of the problem space, some of which are in the human's working memory and some of which are externally perceivable. The potential actions are working memory changes or operators as described above.

Clearly, novel fault diagnosis is a special case of problem solving. The specializations are as follows. First, the human operator must realize the presence of a novel rather than routine failure. Ideally, the displays that result from a novel fault would be sufficiently different from the displays of routine faults. If the novel fault had a display different from routine faults, detection of a novel fault would seem to be assured. Unfortunately, no existing system has been designed from this perspective.

Another specialization is-that novel fault diagnosis will occur when the operator has a problem space designed for routine operations and routine failures. It is not known if an existing problem space representation will interfere with novel fault diagnosis. It would seem difficult to believe that some interference does not occur.

A final distinction between novel fault diagnosis and most problem solving research has been how clearly the human can observe the system and the consequences of changes to it. For example, in cryptarithmetic, the human has complete information about the system, the legal operations, and their immediate consequences. Typically, when an operator controls a complex system, the system state is less clearly perceived, the available operations are larger in number, and their effects less clearly perceivable. The consequences of this imprecision are not well understood.

THE SYSTEM AND THE TASK

The Orbital Refueling System (ORS), a NASA-designed payload on the Space Shuttle, was selected for study [NASA 1985]. The function of the ORS is to refuel orbiting satellites with hydrazine, with the objective of extending their useful service life. As shown in Figure 1, the ORS fluid system contains a variety of components such as tanks, valves, pipes, etc. The operator controls the simulated ORS by opening and closing valves. Transfering fuel from propellant tank 1 to propellant tank 2 might proceed as follows. First, tank 2 pressure is reduced by momentarily opening valves 10, 11, 13, and 17. Second, tank 1 is pressurized by opening valves 1, 3, and 7. Gaseous nitrogen will flow out of the two small supply tanks, be pressure regulated, and fill tank 1 on one side of the bladder. To transfer fuel to tank 2, valves 5, 14, 15, 16, and 9 would be opened. Because this version of the ORS was for demonstration purposes, all transfers take place between the two large tanks rather than to a satellite fuel tank. There are several assemblies whose purpose was not explained in the above example. The relief valves RV1 and RV2 serve as a safety pressure relief. Check valve CV1 prevents backflow into the gas system. The bladders in tank 1 and 2 serve to isolate the fuel from the propellant and also to contain the fuel in the weightlessness of space. Some components (e.g., valves 10 and 11) may seem redundant; they are so by design for two failure tolerance.

The Diagnosis Task

The operator's task is to diagnose the failure in the system. This requires the operator to manipulate and observe the system, because a diagnosis cannot be determined uniquely from an observation of a state vector at a single point in time. A solution is an assignment of states to components such that the assignment's behavior is always identical to system behavior. For a single valve failure, the solution would be a normal state for all components save the failed valve, which might be jammed shut. The diagnosis problem can be viewed as a combinatorial search for a state assignment. The search is constrained by the laws of component physics. That is, a state assignment to a component imposes constraints on its neighboring components. For example, if a valve is opened and permits a flow down a pipe, the component receiving the flow must be in a state to accept the flow.

QUALITATIVE MODELS OF CONTINUOUS PHYSICAL PROCESSES

This section describes qualitative models: representations, the computational problems solved, and the specific needs of our aid of the qualitative model.

A qualitative model is a symbolic representation of a system. Its most basic description is of a component. A component is described in terms of its connections to other components and its behavior. Behavior is described in terms of the physical variables which are present at its connections. The differentiation between the structural description (connections) and the behavioral description is particularly important for insuring the robustness of a qualitative model. The isolation of each component in the behavioral description has usually been emphasized by other qualitative modeling [De Kleer and Brown 1983]. Our qualitative model represents the system at both the component level and at an aggregated level as paths. The motivation for this is the belief that a multi-level description is closer to the operator's internal model of the process.

From a given state, the behavior of a component is described in terms of the physical variables present at its ports. A physical variable (and its time derivative) may take several values. The time derivative usually has only one of three possible values: negative, zero, or positive. The variable itself may take either nominal or ordinal values. The nominal values usually correspond to points at which behavior (component or material) changes. For example, water temperature would have nominal values at freezing and boiling. Variables may also take on ordinal values (or relationships). For example, water temperature could be taken to be greater than freezing and less than boiling.

The nominal and ordinal values taken by physical variables are said to occur in a quantity space [Forbus 1984, Kuipers 1984]. The quantity space is a partial ordering on the physical variable values it contains. The partial ordering occurs because not all comparisons are relevant to understanding the physical system qualitatively. For example, consider a valve between two tanks, A and B. When the valve is opened, the resulting behavior is determined by the pressures in two tanks. The pressure at other unconnected points in the system is unrelated to the above behavior.

One question that is often raised is why bother with qualitative models. They are not, as it turns out, particularly fast or accurate. For engineering purposes they are inferior to analytic or numerical models. The answer to this question is, first, that the aid does not require a qualitative model; any system model will be acceptable if it can provide the required information to the operator. Our motivation for using a qualitative model is to test the hypothesis that humans use such models internally. Obviously, it is difficult to test this hypothesis directly. A weak test would be whether the qualitative model really aids human performance as described here. A stronger test would be finding similar reasoning weaknesses. As mentioned earlier, a qualitative model cannot answer some questions. If well-trained operators could not answer such questions, did not ask such questions, or could not use answers to such questions, there would be evidence for the hypothesis.

AN EXPLORATORY EXPERIMENT

An exploratory experiment was conducted to observe the strategies subjects used to diagnose the ORS. Three Georgia Tech students were used as subjects. The use of college students is usually considered a compromise in experimental research. Since some space shuttle astronauts have been engineers, this compromise is reasonable in this situation.

The training contained both theoretical and practical elements. First, the basics of gas and fluid transfer were reviewed. Second, there was an explanation of the normal and malfunction behavior of each component. Third, subjects were told how to test for a failed component and how to operate the system.

The subjects then solved five single failure malfunctions. The failures were as follows:

(1) Valve 13 leaked, allowing an unexpected pressure drop.

(2) Pressure transducer 2 was biased high.

(3) A leak to the environment developed between valve 10 and 11.

(4) The relief valve was open during a fuel transfer.

(5) Valve 8 leaked.

The data collected included a time-stamped record of the ORS commands issued and a tape recording of the subject´s verbal protocols. The time to solution is shown in Table 1.

Subject Problem	A	B	C
1	28.6	14.4	31.1
2	*13.8	*21.9	3.6
3	13.4	7.9	6.2
4	12.7	10.0	*21.9
5	7.5	8.3	12.3

Table 1. Time to solution. * denotes giving up.

A Post-hoc Analysis of Performance Data

The data from our preliminary experiment suggest several interesting characteristics of human diagnosis behavior, and which in turn suggested some directions for computer aiding. First, the time spent for a successful diagnosis is strongly related with the number of information gathering actions (IGA) (r = 0.79) and the average time between actions (r = 0.77). The latter two variables were not strongly correlated (r = 0.21). The implication of this is reducing the number of information gathering actions (IGA) is an important goal for improving diagnostic performance.

Second, we classified IGA´s into effective ones (EIGA), which reduced the size of feasible hypothesis set, and ineffective ones (IIGA), which did not. We found that the number of EIGA is invariant among subjects and is also not significantly correlated with the total number of IGA. The total number of IGA is correlated with IIGA (corr.= 0.98), which outnumbered EIGA by 2.5 : 1. This suggests that a problem is solved by collecting the right number of EIGA (largely determined by the complexity of the problem). A better performance is possible when the effective actions are executed earlier in the diagnosis.

Third, we investigated how well the subjects detect the abnormal behavior of the system. We assessed the delay in diagnosis due to failures to collect information that would have revealed the abnormal system behavior. The delay showed high correlation (r = .79) with the number of ineffective actions. Also, 75% of effective actions were of abnormal behavior, and the remaining 25% were of normal behavior (negative evidence). Observations on abnormal behavior, if they are correctly interpreted, became effective actions in almost all cases. Thus, abnormal behavior of the system is probably the most important source of effective information.

The conclusion is that, to help the diagnosis, the cues for effective actions need to be given. Abnormal system behavior is worth watching for this purpose. When desiging an aid, a major advantage of using abnormal behavior is that inferring or requesting the human's current hypothesis is not necessary.

Observation of Strategies

There appeared to be three strategies that subjects used: hypothesis-driven evaluation, data-driven evaluation, and topographic search. Hypothesis-driven evaluation starts with the planning of a test procedure for a given hypothesis. The hypothesis needs to be explicit enough to enable the prediction of its resulting system behavior. A test plan would be diagnostic if, given that the hypothesis is true, the response of the system to the test is unique to the hypothesis. When a sufficiently diagnostic test has been planned, the test is executed and its result evaluated. This evaluation tends to be short because it has already been determined what the results might be.

With data-driven evaluation, the subject first examines a piece of data to determine if it is worth closer attention. This examination is done by comparing the data to expected system behavior. If the data turns out to be unexpected (i.e., not explained in terms of previously observed symptoms or normal behavior), then hypotheses are formulated to explain the data. Whether the formulation is successful or not, this piece of data is remembered by the diagnoses as another symptom to be used later during diagnosis.

Topographic search seems to help reduce the mental workload in diagnosis. Both above evaluation strategies involve deep reasoning with functional causalities. With deep reasoning, the former deduces necessary data from a given hypothesis while the latter formulate and evaluate hypotheses from the given data. Topographic search [Rasmussen 1984], without such a deeply based hypothesis, is used to find data. For instance, the sensor near the suspected component are read in hope that the reading may give some diagnostic information. An example of topographic search of hypotheses is suspecting nearby components when a sensor reading is out of the normal range. The differentiation of a single general hypothesis to several more specific hypotheses can be considered as topographic search.

Although it is not relevant to our diagnostic task, other forms of rules may be used as alternative ways of causal search. With experience or specific system knowledge, it is possible to connect a hypothesis with data through function-based reasoning [Rasmussen 1984].

AIDING WITH A QUALITATIVE MODEL

This section describes how the qualitative model is used as a foundation for aiding. For simplicity, the interface will be used to organize the presentation. The interface has four windows: schematic, interaction, sensor display, and hypotheses (Figure 2). Each window will be described first. The types of aiding that occur within the window will then be described. Finally, the justification for the aid, which is the human decision-making suboptimality we hope to mitigate, will be presented. It is possible that a form of aiding and a justification for aiding may apply to more than one window.

Schematic Window

The schematic window displays a schematic diagram of the ORS. The schematic always shows the commanded state of the valves. One form of aiding employed here is the set of components that should be at equal pressure given the commanded valve positions. Whenever the operator opens or closes a valve, the display changes the path to show this property.

The motivation for this is that the operator frequently makes a test among a set of components that should be at equal pressure. It should be noted that the qualitative model uses this same information internally in its simulation of the ORS. A related form of topographic information is flow paths, which are paths that should contain flow if the valves obey their commands.

Both of these forms of aiding support the operator during topographic search [Rasmussen 1985]. From a cognitive standpoint, both aids should lessen working memory loads. It is by no means difficult to determine equal pressure and flow paths without the aid, but it is extra work for the operator to do so.

The second aid and perhaps the most interesting is the what- if model with which the operator may test a hypothesis. The what-if model is a model that is parallel to the system model. The component states of the what-if model are set by the operator. Recall that the diagnosis task is to determine the states of the system components. The operator may use the what-if model to test a hypothesis. For example, suppose valve 13 is hypothesized to be leading. Then, the operator may turn on the what-if model, set its valve 13 to leaking and all other components to normal. When activated, the behavior of the what-if model and simulation are displayed in parallel. The system can be put through a series of state changes to determine if the two behaviors are equal.

The motivation for this aid is to help the operator's mental model of the system. There are two ways this might help. First, the operator may have an incorrect or incomplete mental model. Second, the operator may have difficulty integrating correct component behavior to correct system behavior because of working memory limitations. In either case, the what-if model serves as a substitute for the operator's model. This does not mean that the operator need not understand the system at all; he or she must still set the component state. It also does not mean that the operator may not have trouble using this aid. We will return to this question later.

Interaction Window

The interaction window is where the operator's commands are echoed by the interface. The commands available to the operator include the following:

(1) Opening and closing valves.

(2) Comparing two pressures. On a real physical system, the numerical pressure could be displayed on the schematic. When a qualitative model is used, there is no scale in general to which a pressure can be referred. Instead, a pressure can be referred to other pressures in the system by the relations less-than, equal-to, or greater-than.

(3) Display of the first derivative of pressure (positive, zero, or negative).

(4) Turning the what-if model on and off.

(5) Making state assumptions in the what-if model.

When the what-if model is on, the open, close, and comparison commands apply both to the system and the what-if model.

Sensor Display

The sensor display contains the output from the comparison command: the relationship between two pressures or the first derivative of a pressure. The what-if model, if activated, has its corresponding output displayed side-by-side with the system model.

The aiding that occurs through this window is to indicate which observed behaviors deviate from normal behavior of the system. The aid runs a normal model (that is, a qualitative model with all component states normal) and compares its behavior to the system's behavior. Differences are highlighted. This display differs from conventional warning systems (for example, annunciator panels in nuclear power plants) in that reference is made to a system model, not a fixed point.

The strategy supported by this display is data-driven search, which was observed in our preliminary experiment. In the initial stages of diagnosis, the operator did not have a specific hypothesis. Instead, he or she collected data to develop one. The purpose of this aiding feature is to direct the operator toward more relevant data.

The human decision-making biases that we hope to mitigate all deal with suboptimal use of data or cues. Human have a limited ability to integrate more than three sources of information. Further, humans sometimes use irrelevant data, especially if it is salient. This display attempts to mitigate this by making important differences salient. Another deficiency of humans is a narrow focus of attention. The aid should work against this by displaying all differences, not just those on which the operator has focused.

Hypotheses Window

The hypotheses window will display a set of hypotheses that might be the cause of the observed symptoms. These hypotheses are simply state assignments to components (e.g., valve 13: leaking). The hypotheses will be listed in order of plausibility, according to a heuristic of symptom covering.

Many decision-making biases exist with respect to hypotheses. The one that is directly addressed is the difficulty humans have in generating a complete set of hypotheses [Mehle 1982].

Representativeness, anchoring, and confirmation bias often occur when humans select and evaluate biases. Representativeness refers to the tendency to select hypotheses that are easily recalled from memory. This could be due either to recent use of the hypothesis or to a close match between actual symptoms and symptoms covered by the hypothesis. Anchoring refers to the tendency to stay with an initial hypothesis even after it has been disconfirmed. Confirmation bias is the tendency to test data that will only confirm a hypothesis. It is in effect a failure to seek negative evidence. To mitigate these biases requires meta-aiding, as described below.

Meta-aiding

Earlier, we mentioned that the operator may have difficulty using the what-if model. Recall that the operator must make assumptions about the states of components. Having a what-if model means the evaluation of assumptions is easy, but making assumptions is not aided by the what-if model.

Meta-aiding is aiding the use of the what-if model--specifically, helping the operator choose component state hypotheses. While listing these hypotheses in the hypotheses window is an aid, it may be necessary for the interface to take a more active role. If anchoring and confirmation bias occur, it will be necessary for the interface to determine when the operator's hypothesis (expressed in the what-if model states) is no longer valid. When this occurs, the interface will step in to warn the user of his or her mistake.

CONCLUSION

An aid has been described for novel fault diagnosis in complex systems. To the best of our knowledge, this aid is unique in the following ways. First, the emphasis is on novel rather than routine faults. Second, it contains a qualitative model that may correspond to the human's internal model of the system. This model represents knowledge only of how the system works. Many of the proposed aiding schemes are proceduralized fault finders: they tell the operator what action to take. Third, the qualitative model is the basis for much of the aiding that takes place. Fourth, the interface specifically attempts to mitigate some human decision-making suboptimalities during fault diagnosis.

The current status of this aid is as follows. The aiding software for topographic path displays, flow paths, and the what-if model have been implemented. Hypothesis generation and the corresponding suboptimality detection have not. We feel it is premature to implement suboptimality detection (i.e., meta-aiding) without some experience with aiding by topographic displays and the what-if model.

ACKNOWLEDGMENT

This research was supported by NASA-Ames Research Grant NAG 2-123. Dr. Everett Palmer is the technical monitor.

REFERENCES

Davis, R., "Diagnostic reasoning based on structure and behavior," Artificial Intelligence, Vol. 24, pp. 347-410, 1984.

De Kleer, J. and Brown, J.S., "Assumptions and ambiguities in mechanistic mental models," in D. Gentner and A.L. Stevens (Eds.), Mental Models, Hillsdale, NJ: Lawrence Erlbaum Assoc., 1983.

Forbus, K., "Qualitative process theory," Artificial Intelligence, Vol. 24, pp. 85-168, 1984.

Gentner, D. and Stevens, A.L. (Eds.), Mental Models, Hillsdale, NJ: Lawrence Erlbaum Assoc., 1983.

Kuipers, B., "Commonsense reasoning about causality: Deriving behavior from structure," Artificial Intelligence, Vol. 24, pp. 169-203, 1984.

Mehle, T., "Hypothesis generation in an automotive malfunction inference task," *Acta Psychologica*, Vol. 52, pp. 87-106, 1982.

Miller, R.A., Pople, H.E. Jr., and Myers, J.D., "INTERNIST-1: An experimental computer-based diagnostic consultant for general internal medicine," in *Readings in Medical Artificial Intelligence*, Clancey, W.J. and Shortliffe, E.H. (eds.), Reading, MA: Addison-Wesley, 1984.

Morris, N.M. and Rouse, W.B., "The effects of type of knowledge upon human problem solving in a process control task," *IEEE Trans. Systems, Man, and Cybernetics*, Vol. SMC-15, No. 6, 1985.

Morris, N.M. and Rouse, W.B., "Review and evaluation of empirical research in troubleshooting," *Human Factors*, Vol. 27, No. 5, October 1985.

NASA Johnson Space Center, "Orbital refueling demonstration system description," Program Development Office, October 21, 1985.

Newell, A. and Simon, H.A., *Human Problem Solving*, Englewood Cliffs, NJ: Prentice-Hall, 1972.

Rasmussen, J., "Skills, rules, and knowledge: signals, signs, and symbols, and other distinctions in human performance models," *IEEE Trans. Systems, Man, and Cybernetics*, Vol. SMC-13, No. 3, pp. 257-266, 1983.

Rasmussen, J., "Strategies of state identification and diagnosis in supervisory control tasks, and design of computer based support systems," in W. Rouse (Ed.), *Advances in man-machine systems research*, Vol. 1, JAI Press, 1984.

Rasmussen, J., "The role of hierarchical knowledge representation in decision making and system management," *IEEE Trans. Systems, Man, and Cybernetics*, Vol. SMC-15, No. 2, pp. 234-243, 1985.

Rouse, W.B. and Morris, N.M., "On looking into the black box: Prospects and limits in the search for mental models," to appear in *Psychological Bulletin*, 1986.

Shepherd, A., Marshall, E.C., Turner, A., and Duncan, K.D., "Diagnosis of plant failure from a control panel: A comparison of three training methods," *Ergonomics*, Vol. 20, No. 4, pp. 347-361, 1977.

Shortliffe, E.H., *Computer-Based Medical Consultation: MYCIN*, New York: American Elsevier, 1976.

Wickens, C.D., *Engineering Psychology and Human Performance*, Columbus, OH: Charles E. Merrill, 1984.

Wohl, J.G. "Maintainability prediction revisited: Diagnostic behavior, system complexity, and repair time," *IEEE Trans. Systems, Man, and Cybernetics*, Vol. SMC-12, No. 3, 1982.

Wohl, J.G. "Cognitive capability versus system complexity in electronic maintenance," *IEEE Trans. Systems, Man, and Cybernetics*, Vol. SMC-13, No. 4, 1983.

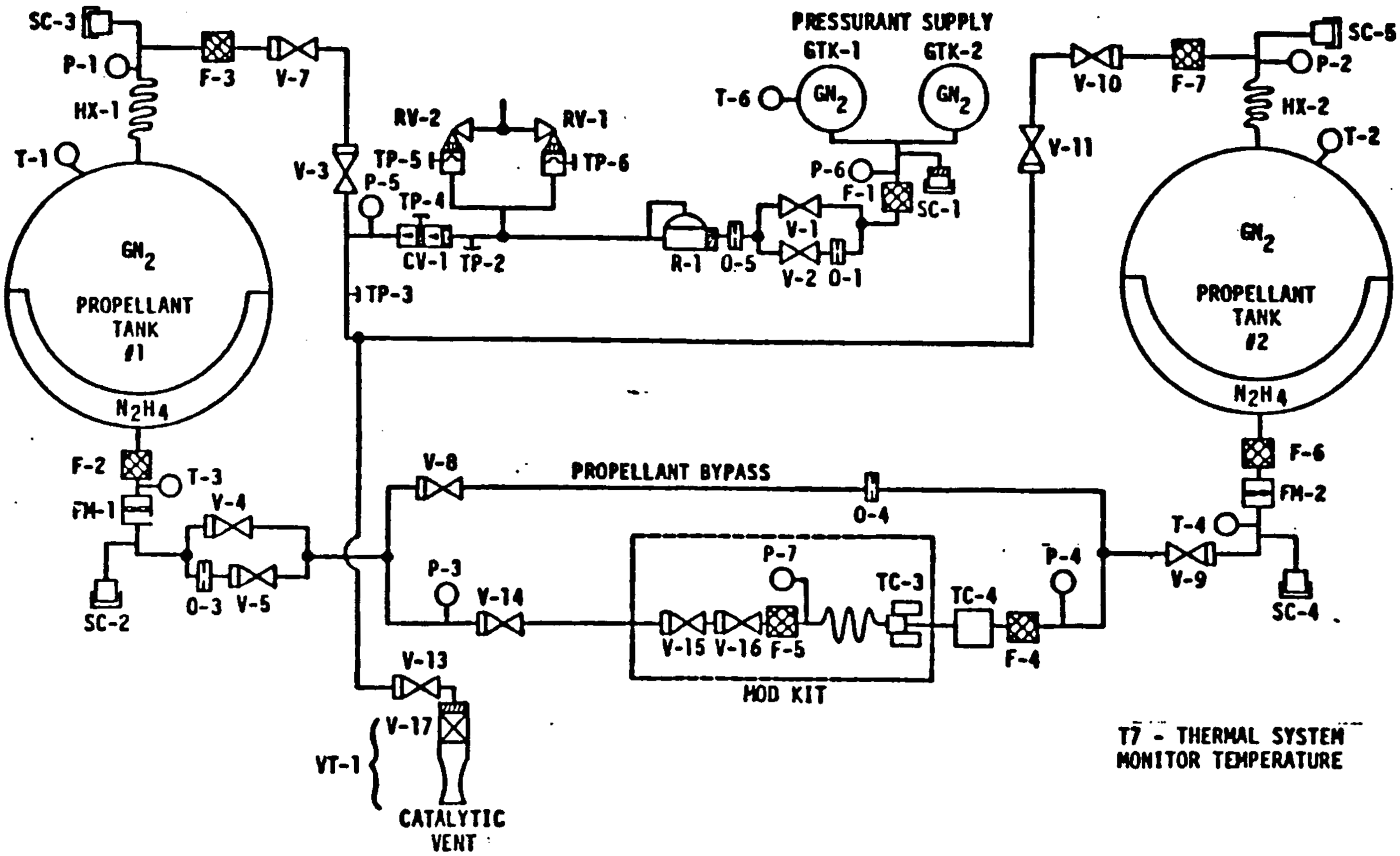


Figure 1. The Orbital Refueling System.

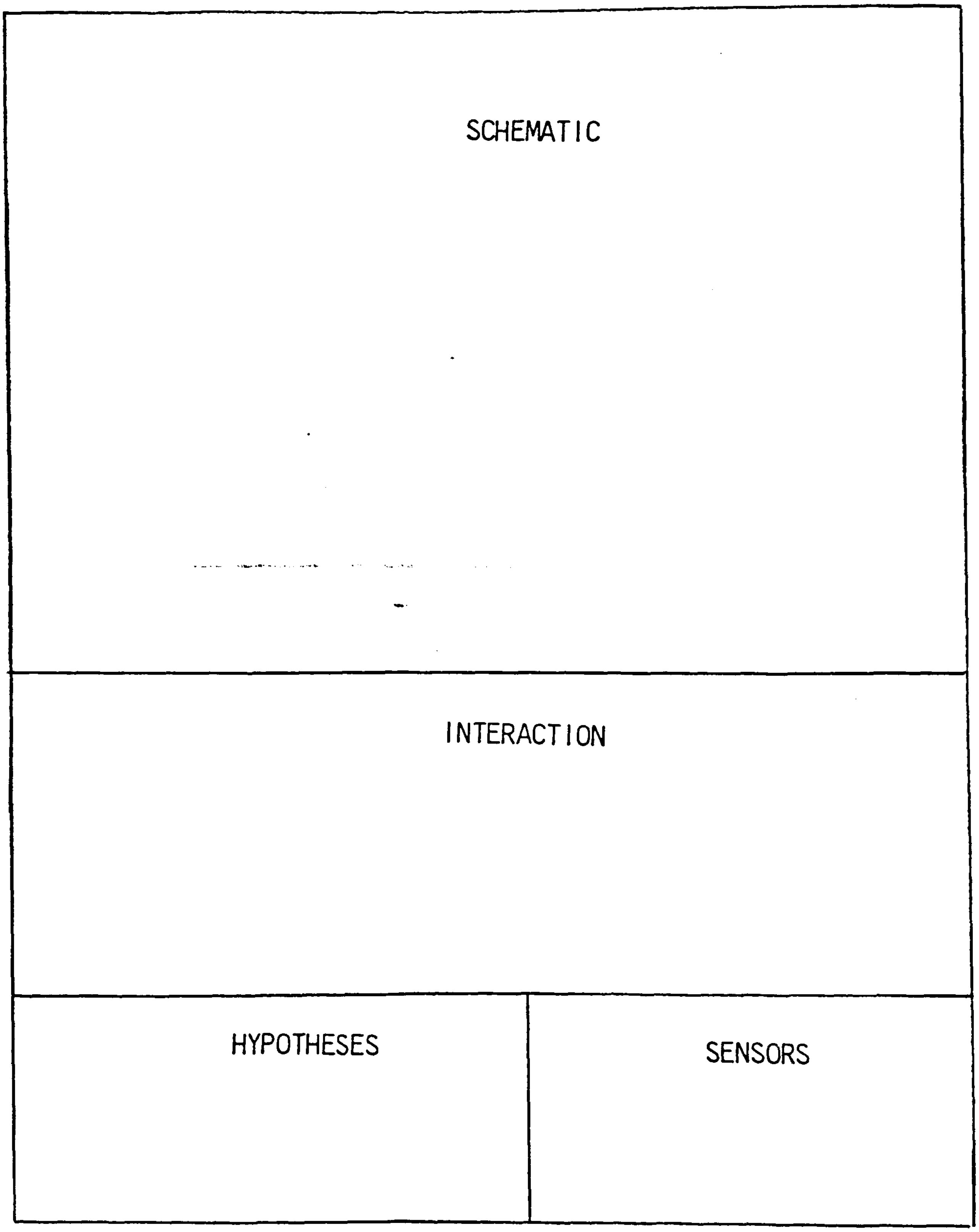


Figure 2. The operator's display.

C

```
               C36
  <-VT----V17-------V13                RV     C33    C32        C31  /""""""\
                       | C35           |             .-----V1---.    | GTK |
    .--V7----V3---------------< CV <-------REG--05--:            :---\______/
    |    C37       |       |          C34            `---V2--01-'        |
    |C38           |       @ P5                                          @ P6
    |              |                       C39          C40
/""""""""""\~@ P1  `------------------V11------V10---------------/""""""""""\~@ P2
| TK1G/L |                                                       | TK2G/L |
\_________/                                                      \_________/
    |                          @ P3              @ P7    @ P4             |
    | C1                       |     C6    C7    |       |         C10|
    |        .-----V4---. C4 .----V14---V15---V16-->TC<-----.         |
    `-------:           :----|                   C8        C9|----V9--'
             `-03----V5-'    `-------V8--------04-----------'
                                          C5
```

Exercise 1.
Situation: P1 is found too low and still decreasing.
Fault: V13 leak
valves open: V3, V7, V10, V17 / V4, V14, V16, V9

Exercise 2.
Situation: P2 appears to be too high.
Fault: P2 high bias
valves open: V1, V3, V10, V17 / V5, V15, V9

Exercise 3. (confg-1)
Situation: P1 is low and decreasing.
Fault: pipe leak between V10 and V11 (c39)
valves open: V3, V7, V13, V11, / V4, V14, V15, V9

Problem 1.
Situation: During a fuel transfer TK1L -> TK2L,
P2 does not increase.
Fault: V5 fail closed
valves open: V3, V10, V11, V17 / V5, V14, V15, V16, V9

Problem 2.
Situation: P2 is too high. V11 was found leaking, but
there is one more anomaly.
Fault: V7 failed open
valves open: V3, V13, V10 / V4, V14, V16, V9

```
  <-VT----V17-------V13               RV                           /"""""\
                       |               |             .-----V1---.   | GTK |
    .--V7----V3---------------< CV <-------REG--05--:           :---\_____/
    |             |       |                          `---V2--01-'       |
    |             |       @ P5                                         @ P6
    |             |
/""""""""\~@ P1   `------------------V11------V10---------------/""""""""\~@ P2
| TK1G/L |                                                      | TK2G/L |
\________/                                                      \________/
    |                         @ P3              @ P7     @ P4            |
    |                         |                 |        |               |
    |       .-----V4---.    .----V14---V15---V16-->TC<-----.         |
    `-------:          :----|                              |----V9--'
            `-03----V5-'    `-------V8--------04-----------'
```

Problem 3.
    Situation: P2 is lower than it shoud be.
    Fault: V8 leak (while P2 > P1)
    valves open: V3, V7, V10, V17 / V4, V15, V16, V9

Problem 4. (confg-2)
    Situation: During a fuel transfer TK1L -> TK2L,
                  P2 increases too slow.
    Fault: pipe between V3 and V7 (C37) leaks
    valves open: V3, V13, V11, V10 / V5, V14, V16, V8, V9

Problem 5.
    Situation: a gas transfer GTK -> TK2G proceeds too slow.
    Fault: V15 leak
    valves open: V3, V17, V1, V11, V10 / V4, V14, V16, V9

Problem 6.
    Situation: During a gas transfer from GTK to TK1G,
                  P1 increases too slow.
    Fault: RV leak
    valves open: V1, V3, V7, V10, V13 / V4, V14, V9

B

INSTRUCTIONS - Part 1.

TIME (:   )

I. The Orbital Refueling System (ORS)

The purpose of the ORS is to refuel orbiting satellites on their orbits. As shown in Figure 1 (in the separate sheet), the ORS fluid system contains various components such as tanks, valves, pipes, etc. Because this version of the ORS was for demonstration purposes, all transfers take place between the two tanks rather than to a satellite.

Let's look at the components in the schematic. First, 'XX' and '==' indicate closed and open valves respectively. The operator controls the ORS only by opening and closing valves. For example, You can open/close V3 by the commands OP V3 and CL V3.

There are 4 orifices: namely O1, O3, O4, and O5. Find them in the Figure. An orifice is a designed source of resistance. When there is a mass flow through an orifice, there is a pressure reduction across it. Dropping pressure through orifices is at times useful to control the flow rate. Also, O1 and O5 reduce pressure to the regulator.

Now find GTK, which stands for the Gas TanK. This tank contains high pressure nitrogen gas. Find Tank1 (TK1G and TK1L) and Tank2 (TK2G and TK2L) too. They are the fuel tanks. TK1G is the gas part of Tank1, which is separated by a flexible diaphragm from the liquid part (TK1L) of the tank. The two parts always share the same pressure.

On the path from GTK to TK1G, you will find 'REG' (REGulator) and 'CV' (Check Valve). The regulator produces a constant output gas pressure even though the input pressure varies. The check valve allows the gas to flow forward only (i.e., right to left).

Find 'RV'. It stands for a Relief Valve. If the pressure goes up beyond some dangerous level, the relief valve will automatically open to decrease the pressure. The operator can also manually open/close the 'RV' as any ordinary valves by OP RV or CL RV. At the top left, you see 'VT', which stands for VenT. You may release pressurizing gas through the vent by opening V13 and (___).

The lower half of the schematic (from 'TK1L' to 'TK2L') is the liquid (fuel) part. There, 'TC' is for Terminal Coupling and is assumed always being connected during our diagnostic missions.

To transfer fuel from 'TK1L' to 'TK2L', Tank1 needs to be first pressurized by opening valves betwen GTK and TK1G. In the above schematic, TK1G is being pressurized by the gas through the open valves (___), (___), and (___). Since TK1L has always the same pressure as TK1G, it is being pressurized too. Then, the gas flow may be stopped by CL V2. The fuel may be transferred by opening valves between the two tanks. In the above, the operator would simply open (___), hence issue a command (_________) to do this. The tank of higher pressure will become the source and the other will receive the fuel.

The following is important. There are seven pressure sensors (P1 to p7) in the ORS, (__),(__),(__), and (__) in the gas part, and (__),(__), and (__) in the liquid part. To read them, you have only two commands:

D P1

: to see the 'D'erivative of P1.

Answer: + for P1 increasing, - decreasing, and = constant.

C P2 P4

: to 'C'ompare P2 and P4.

Answer: > when P2 > P4, < when P2 < P4, and = when equal.

The command D is valid only for tank pressures, namely, P1, (___), and (___). In pipes, unlike the tanks which have considerable capacity, the pressure change is instantaneous so that you can't expect to see + or - as the answer to D P5. D IS MOSTLY USED TO CHECK IF THERE IS A FLOW FROM/INTO A TANK.

As the gas or liquid flows from one tank to another, its pressure decreases along the path. A pressure drop can only occur across a resistance. When the fluid passes an orifice, which has significant resistance, the pressure will decrease. An abrupt change in the conduit shape, such as from a pipe to a tank or vice versa, also produces resistance and results in a pressure drop. We will assume that pipes or valves normally have negligible resistance. C is the command which is frequently used to check the pressure drop along the path.

Another use of C command is to check if two sensors, which are supposed to be equal, agree with each other. When two or more sensors are connected by an open path, and if there is no flow through the path between the sensors, they all should read the same pressure. Resistance doesn't matter when there is no flow. If the sensors read differently, either there actually is a flow in the path (e.g., due to a leaky valve or pipe) or at least one of the sensors is wrong.

Keeping this in mind, you are now able to predict how the sensors will behave when you open or close valves. Three situations are summarized here.

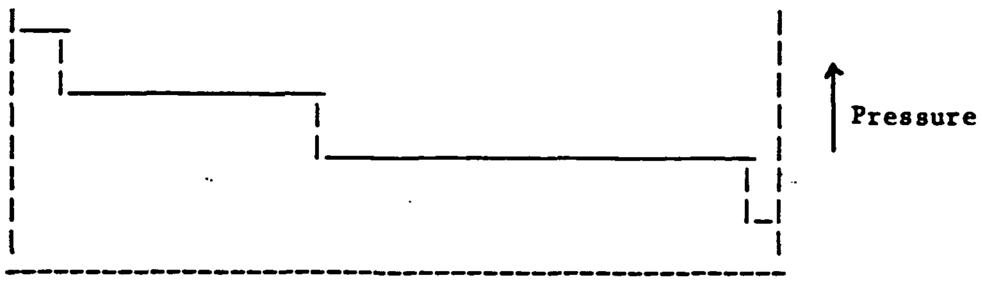
1. When a flow exists.

   a. The pressure decreases in the source and increases in the sink.

b. The pressure drops by resistance while the material travels along the path. This is shown in the following diagram.

```
TANK 1 --------------01---------==12------------TANK 2
(source)                                        (sink)
```

Flow ----->



In Figure 1, there is a gas flow from GTK to TK1G.

Predict the results:

D P6 --> (__)

D P1 --> (__)

C P6 P5 --> (__)

C P5 P1 --> (__)

Now, how about

C P1 P3 --> (__) ?

Did you consider that an orifice reduces pressure only when there is a flow through it? If not, check your answer again.

Now, suppose P2 > P1 and V8 is open.

Then the fuel will flow from (____) to (____) and:

C P2 P4 --> (__)

C P4 p7 --> (__)

C P4 P3 --> (__)

C P3 P1 --> (__)

2. In case that there is no flow (V8 is closed again).

In Figure 1, there is no flow from or to 'TK2G' and 'TK2L'.

All the pipes around the tank will share the same pressure.

Thus,

C P2 (__) --> =

C P2 (__) --> =

If you close V9, the pressures on both sides (will, will not) change. Therefore,

C P2 P7 --> (__)

On the other hand, when you close v3, you expect

C P5 (__) --> =

3. Special case of 2.

In Figure 1, suppose V17 is leaking.

It is open to the environment which has zero pressure.

Even though the operator closes V3, the gas will continue to flow from (_____) to the environment. Thus,

D (__) --> -

C P5 P6 --> (__)

C P5 0 --> (__)

If the operator closes V2, Since the pipes do not have significant capacity, the gas escapes right away. Therefore, immediately after closing V2, you get

C P5 0 --> (__)

** We assume the capacity of components (except tanks) to be always negligible however small the size of a leak may be.

The same will be (true, false) when V17 is closed but the pipe between V13 and V17 leaks.

TIME (:   )

BEFORE YOU START NEXT PART, RETURN THIS PART TO THE EXPERIMENTER.

TIME (:)

II. Malfunctions

We will now discuss possible malfunctions for each component.

1. valve (including check valve and relief valve)

   a. leak - in spite of being commanded to be 'closed', it allows some, though not a full, flow. There is a resistance when commanded closed. When commanded open, it acts normally.

   b. fail open - no matter what you command, it remains fully open.

   c. fail closed - no matter what you command, it remains closed.

2. regulator

   a. fail open - always remains fully open without reducing the pressure.

   b. fail closed - always remains closed whatever the input pressure is. (No gas passes through the regulator.)

3. orifice

   a. fail open - fails to provide resistance or pressure drop, allowing the material to flow freely.

   b. fail closed - prohibits flow.

4. conduit (including 'TC', the terminal coupling)

   a. leak - leaks gas or liquid to environmental space. (remember that a valve leak is THROUGH the valve, not to the environment)

   b. fail closed - completely prohibits flow.

5. vent

Since the ´VT´ is a simple opening to external space, its working and malfunctioning is the same as a conduit.

6. sensor

a. biased high - reports a higher pressure than the actual one.

b. biased low - reports a lower pressure than the actual one.

c. dead - fails to follow the change of pressure, reading 0 or other fixed pressure.

TIME (:   )

III. Commands

We will summarize the commands you can use. There are only two commands for operation -- OP and CL. You can open or close only the valves. Examples are

```
OP V3
CL V17
```

There are two commands, ´C´ and ´D´, to get information about pressure through the sensors. Followings are the examples.

```
C P1 P3
D P2
C P5 0
```

The last command compares P5 with 0, the environment pressure of outer space.

** Now, call the experimenter. You may ask him any questions.

TIME (:   )

IV. An example of ORS operation.

** You need to use the terminal for this section. The experimenter will help you through this section.

Now, you will undertake a very typical operation as an exercise. Also, through this example, you can become more familiar with the commands. Simply follow the steps one by one with care. Don´t open/close the valves otherwise, although you can freely read any sensors at any time.

a. type: (EXERO) and hit ´return´.

The familiar schematic now appears on the top half of the screen. Notice that the symbol ´XX´ indicates a closed valve, and ´==´ an open valve. The symbol shows so-called ´commanded´ position. The actual position can be different from this switch position when a valve malfunctions.

The fuel needs to be transferred from ´TK1L´ to ´TK2L´. To achieve this transfer, the pressure in ´TK1L´ should be higher than that of ´TK2L´. So, let´s pressurize the source tank by providing high pressure from GTK

Please write in your answers whenever you are asked.

b. type: OP V1

What happens? (When you are asked like this, write down your guess on the system behavior resulted by the command.)

Try to confirm the above answer by observing sensors. Then, give a set of commands (including at least a 'C' command) that are useful for this.

c. type: CL V1

What happens?

How do you confirm it? (answer as in b.)

d. type: OP V8

What happens?

How do you confirm it? (answer as in b.)

Check 'C P3 P4', 'C P4 P7' and 'C P1 P5'. Can you explain them?

e. type: CL V8

OP V16

Check ´C P3 P4´. Can you explain it?

e. type: CL V4

Give all the sets of equal pressure sensors.

TIME (:)

Congratulations! Your first mission has successfully been completed.

INSTRUCTIONS - Part 2.

Before you start, please review Part 1 again. Especially, you need to be familiar with sections II and III of Part 1.

I. Diagnoses

The followings are examples of typical diagnostic procedures. Following the reasoning, fill in the parentheses.

a. To check a sensor

See Figure 1. Suppose you want to check the sensor P3. You can close V3 and expect (___) to read the same as P3. If not, P3 probably is bad. Of course, the bad one may be (___) rather than P3. To check further, you can close V9, open (___), and compare P3 to P4 or (___).

b. To check a conduit leak (to environment)

If there is a leak between V4 and V14, D P1 can either give - or + depending on whether the input flow rate to TK1G is greater than the output rate from TK1L. If you close V7, a leak between V4 and V14 will cause a decrease in the sensor (___). But, when the valve (___) is closed, TK1L will stop loosing the pressure. This means that the leak is in the { left, right } hand side of the valve. Another evidence of a leak between V4 and V14 is that C P3 0 ---> = { before, after } you close V5.

c. To check a valve leak

Suppose you found D P2 gave +. This is possible if one of (___) and (___) is leaking. You may suspect that even two valves (___) and (___) failed together. If you close V10 and find the flow stopping, which makes (_____) return =, you have the evidence that the flow was from (____) and the leaky valve was (___).

If closing V10 does not stop the flow, you will first suspect (___) since one valve failure is more likely than a two valve failure. If closing V5 or V9 results in D P2 ---> =, the problem is in the { gas, liquid } part. Now, after you open V5 or V9 again, if closing V16 stops the flow, then the flow was through { V8, V14 and V15 }.

Now, let us consider several situations to see how you can test your hypotheses. You will be given a hypothesis for each problem. Each hypothesis implies that only one component is suspected. Prove or disprove the hypothesis.

TIME: (:   )

1. Hypothesis: the pipe between V13 and V17 leaks.

   Type (HYP01) and start when the diagram appears.

2. Hypothesis: V11 leaks.

   Type (HYP02) and start when the diagram appears.

3. Hypothesis: V2 failed closed.

   Type (HYP03) and start when the diagram appears.

4. Hypothesis: CV failed open.

   (Hint: you can open/close RV as well as other Valves.)

   Type (HYP04) and start when the diagram appears.

5. Hypothesis: P2 is biased high.

   Type (HYP05) and start when the diagram appears.

TIME: (:   )

II. Exercises

When you are diagnosing the ORS, you will be introduced to a malfunction situation and given the symptoms so far identified. The previous operation was being done by another personnel. Your mission is to diagnose the system and find out the anomaly AS PRECISELY AS POSSIBLE so that another crew could easily fix it. For example, if you suspect a valve leak, you have to continue until you can say which valve it is. A conduit malfunction can be traced down to ′between valve a and valve b′, where valves include the check valve (CV).

You have to THINK ALOUD during the diagnosis. That means, you should utter everything that arises in your mind or in action. DON′T try to EXPLAIN what you HAVE thought; speak out WHILE you are THINKING. Speaking must not be an extra work. You don′t have to give complete or composed sentences. The components which have names on the schematic may best be called by the names. Others, mostly pipes, may easily be called ′right to′ or ′left to′ a named component. Again, please KEEP TALKING OUT. Speak everything that goes on in your mind regardless of its importance. Also, whatever you type in on the keyboard needs to be spoken out. If you stop speaking for any length of time, the experimenter will prompt you with "What are you thinking?"

Your performance is measured by the sum of time you spend for the problems; solve the problems in as little time as possible. However, give your answer only when you are completely convinced it is correct. And, don′t give up, at least easily. The penalty for a wrong answer is great; giving up, even greater.

Now, proceed with exercises 1 and 2.

RETHINKING EXPERIMENTAL PROCEDURE

Findings from the 1st Experiment (Testing N Feature)

1. With enough training, the problem complexity becomes the biggest source of variation.
2. Subject variation may be reduced as much as to a standard deviation of around 0.3 mean.
3. The training effect was examined using Time/IGA. It was quite stable and showed similar pattern from problem to problem among subjects.
4. No significant interaction between the training effect and the aiding effect or subject effect were indicated from the data.
5. The "N" feature did not show positive effects.

Refinement of the Training Procedure

1. More exercise (2~3) problems are needed for "warming-up" before the actual problems.
2. Clearer statements and no question for the 1st session and "solve-it-together" for the 2nd session.

Experimental Design

1. The constraint of having to give a problem to a subject only once restricts the possibility of a factorial design. No replication in the S X P cells leaves the two following designs.
2. Design 1 confounds Problem and Position.

   Design 2 is a Graeco-Latin design which separates Problem and Position.

	P1	P2	P3
S1	ua	0	0-N
S2	0	0-N	ua
S3	0-N	ua	0

Design 1.

	1	2	3
S1	P1,ua	P2,0	P3,0-N
S2	P3,0	P1,0-N	P2,ua
S3	P2,0-N	P3,ua	P1,0

Design 2.

3. Confounding Problem and Position

- As long as the training effect is not correlated with the aiding effect, this design will not degrade the efficiency or validity of the experiment. (We try to minimize the training effect, anyway.)
- Although the training effect is not measured separately, it is not an important purpose of this experiment.
- This design allows freedom of replication and keeps the analysis relatively easy.

4. Graeco-Latin Design

- The main advantage is that we may estimate the training effect. However, the training effect is closely related to the problems. There would be more learning from a difficult, hence long, problem. If such a problem comes first, more improvement will occur after the first session. This violates the no-interaction assumption in a Graeco-Latin Design. Not only the training effect will not be properly estimated, but also the efficiency of test will be degraded since the actual interaction will be merged to the error term.
- Design 1 allows more flexibility of replication. 9X6 or 12X6 are possible replications with Design 1, but are not allowed in Design 2.

5. Conclusion

- If we are concerned with the Training effect, than we need to confound it with Problem since there may be a strong interaction between the two. If the Training effect is not so high (which is the likelier case as the data indicates), Design 1 is readily justified.
- To estimate the interaction between Problem and Aiding, we need replication with subjects for each treatment combinations. This leads to the following design (Winer, "Statistical Principles in Experimental Design", 1962).

	P1	P2	P3	P4	P5	P6
G1	-	0	0-N	-	0-N	0
G2	0	0-N	-	0	-	0-N
G3	0-N	-	0	0-N	0	-

In this plan, G1, G2, and G3 are groups of an equal number of subjects. If the interactions with the group factor are negligible (this assumption is reasonable if the groups represent random subsamples from a common population),
the following model will be appropriate for the analysis (Winer, 1962).

$$E\ [Y(ijkm)] = m + G(k) + S(k|m) + P(i) + A(j) + P.A(i,j)$$

where $G(k)$ is the effects associated with groups and $S(k|m)$ effects associated with subjects within the groups.

EXPERIMENTAL PROCEDURE

I. Purpose of the Experiment

There are diagnostic situations in which causal reasoning about the physical system plays a central role. Such situations may be created by a system failure that the operator has not experienced. The irrelevancy of previous experience prohibits a direct mapping from symptoms to causes. Also, the base rates for hypotheses are normally not available due to the lack of experience. As a result, the diagnosis will primarily be based on causal reasoning about the system.

Aiding based on a qualitative model of the system seems to deserve consideration because the human's causal reasoning is also claimed to be qualitative. The qualitative model will be able to predict and describe the system events which are believed to be important to human reasoning. This should cause the information produced by the model to be highly compatible with the human information processing.

One purpose of this experiment is to test the validity of this aiding approach. More detailed interest is in the relative effectiveness of different aiding information that can be provided by the model. In the next section, the experiment planned for this purpose is described. The design of experiment and the analysis of results are discussed in the last section.

II. The Experiment

This section begins with a brief description of the Orbital Refueling System (ORS), which is the context of problem solving in the experiments, and the interface. A more detailed discussion may be found in the previous papers

[Proceedings of the 1986 IEEE International Conference on Systems, Man, and Cybernetics, pp.1222-1227; IEEE Transactions on Systems, Man, and Cybernetics, to appear] and the thesis proposal. Then, a description of the experiment in terms of problems, independent and dependent variables, subjects, and training will follow.

The ORS and the Interface

In the ORS as described in the thesis proposal, as in most plants, it is not possible to test each component directly. A diagnostic hypothesis can only be examined indirectly through testing operations. Because of this, the diagnosis of a novel failure in this system will more heavily rely on causal reasoning. This makes the ORS a good problem solving context for our experiment.

The ORS is qualitatively simulated on the center´s Vax 11/780 computer. The interface has four windows (Figure 1). The schematic window shows a schematic diagram of the ORS. The commanded positions of valves are shown on

```
 -------------------------------------------
|                                           |
|                                           |
|           S C H E M A T I C               |
|                                           |
|                                           |
|-------------------------------------------|
| I N T E R A C T I O N                     |
|-------------------------------------------|
|                    |                      |
|   S E N S O R S    | H Y P O T H E S E S  |
|                    |                      |
|                    |                      |
 -------------------------------------------
```

Figure 1. The ORS Interface

the schematic. Below the schematic, the operator's commands are echoed in the interaction window. The operator can open/close valves, read the time derivative of a pressure sensor, and compare two pressure sensors. The output from the above sensor display commands is displayed in the sensor window. Under certain aiding conditions, suggested sensor readings will also be displayed in this window. The hypothesis window is used only with an aiding feature. It displays a set of hypotheses set by the operator. These hypotheses are simply state assignments to components.

Problems

For each problem, the subject is given a detected symptom and asked to diagnose the malfunction as precisely as possible. There may be one or two bad components. When two components are bad, the subject is told of one malfunction and is asked to find the other. The problems include valve leaks, pipe leaks, blocked valves, a check valve failure, a relief vent leak, and sensor failures.

Independent Variables

The effects of different aiding information will be examined. Each type of information correspondents to a hypothesized, model-based processing that the operator does during diagnosis. The first processing is called N, which is to predict the normal system behavior after a given operation. The second is O, which is to envision the actual system behavior from limited observation. The third is O-N, the difference between O and N, which is often crucial in an efficient search for the diagnosis. The last processing is called O-H, which calculates the discrepancies between the observed system response and the operator's hypothesis.

Dependent Variables

Many different performance measures were tried with our data from the pilot experiment. The number of information gathering actions (#IGA) appears to be a clear alternative to the time to solve (TIME). An information gathering action is judged to be effective when it reduces the size of feasible hypothesis set. To achieve this, an IGA should be able to remove at least one hypothesis from the feasible set. In addition, it must not be redundant with respect to the information so far collected. We have denoted the number of effective IGA's by #EIGA, and that of ineffective ones by #IIGA.

The pilot experiment showed that #IIGA is a good predictor of TIME ($r = 0.83$; $p < 0.01$). Although several other measures were examined with the data, they either turned out to have insufficient resolution or showed high correlations with the above measures. Thus, the aboves will be the most important measures in the main experiment. However, other measures will be collected for supplementary analysis. The measures are:

Time :	Time to solve the problem
#IGA :	Total number of Information Gathering Actions
#EIGA:	Number of Effective IGA
#IIGA:	Number of Ineffective IGA
#BT :	Number of Bad Tests of Good Hypotheses
#BH :	Number of Good Tests of Bad Hypotheses
#RT :	Number of Redundant Tests

Subjects

Eighteen to twenty four undergraduates in the ISyE 3010 class will serve as volunteer subjects. The subjects will receive extra credit for participating this experiment. They are motivated by giving different extra credit according to their performance: 7% for top one third of the subjects, 6% for the next one third, 5% for the rest.

Training

The goal of our training is to facilitate the subjects with correct causal reasoning about the ORS and reasonably stabilized diagnostic skills. However, if a subject is exposed to a kind of problem several times in a short period, the subject may develop diagnostic procedures that do not require causal reasoning. That means the problems become routine failures rather than novel ones to the subjects.

Two training sessions will prepare the subjects for the final experiment. Training session 1 starts with basic principles derived from fluid dynamics. Then, possible malfunctions for each component are discussed. Finally, the subjects will undertake a simulated ORS mission, during which envisioning of normal system response is practised. Session 2 teaches elementary diagnostic procedures such as checking a sensor bias or a valve leak. The subject then is required to plan testing procedures for five typical hypotheses. Each procedure will be discussed with the experimenter until the subject develops (and understands) a correct procedure. The subject then solve three real problems as exercises. Session 1 usually takes 1 to 1.5 hours. Session 2 is normally takes 2 hours, but varies depending on the subject′s pace.

The performance of subject in the training sessions is closely monitored. The principles part contains many questions to ascertain proper understanding.

The answers are checked during the same session and, whenever necessary, discussed again. Problem solving exercises are also attended by the experimenter and necessary discussion or re-explanation is provided. The result is that initially poorer subjects will spend more time in training rather than end with poor understanding. Our experience is that by the end of the second session, subjects performed satisfactorily and showed little additional improvement in diagnostic skill.

III. Experimental Design

Rationale for Three Experiments

The features will be examined by three experiments. The display of aiding information constrains those features that can be tested together. A subject should not be exposed to both N and O features since severe interference is expected. This is because O and N information is displayed identically but has different meaning.

O-H and O-N for the same reason should not be used together. When O-H is used, it acts as O-N until the subject expresses one or more hypotheses. This makes a direct comparison between O-N and O-H difficult. Even if O-H really improves the performance, its contribution will be depend on the extent to which a subject uses it. Different performance criteria need to be used to evaluate the potential benefit of O-H. (The frequency of bad hypothesis testing (#BT) should be emphasized rather than time to solve (Time). The ratio of #EIGA and #IIGA with or without a hypothesis selected may also be compared. These comparisons need to be made against the O-N aiding condition.)

The above considerations led to the following three separate experiments.

1. Test of N against unaided situation
2. Test of O, O-N, against unaided situation
3. Test of O-H against O-N

Differences in the complexity of problems and differences between users are expected to introduce large variation in the performance. To enhance the efficiency of the experiment, a Latin square design which uses problem and subject as two blocking variables is desirable. The treatment levels will be counterbalanced for practice effects. Also, the Latin square design may be replicated to attain enough data points. This design is used for all three experiments. The ANOVA table for this design is given in Appendix A. The first experiment to evaluate the N feature is shown in Figure 2. Figure 3 shows the experiment for testing O and O-N features.

The above design does not estimate interactions because only first order effects are of interest. There is no hypothesis that corresponds to an

		PROBLEMS					
		P1	P2	P3	P4	P5	P6
SUBJECTS	S1	N	-	N	-	N	-
	S2	-	N	-	N	-	N
	S3	N	-	N	-	N	-
	S4	-	N	-	N	-	N
	S5	N	-	N	-	N	-
	S6	-	N	-	N	-	N

Figure 2. Latin Square Design for N effects in Experiment 1.

		PROBLEMS					
		P1	P2	P3	P4	P5	P6
	S1	-	0	0-N	-	0-N	0
	S2	0	0-N	-	0	-	0-N
SUBJECTS	S3	0-N	-	0	0-N	0	-
	S4	-	0-N	0	-	0	0-N
	S5	0	-	0-N	0	0-N	-
	S6	0-N	0	-	0-N	-	0

Figure 3. Latin Square Design for 0 and 0-N in Experiment 2.

interaction between 0 and 0-N.

Pairwise comparisons will be executed using procedures by Tukey, Bonferroni, and Scheffe [J. Neter and W. Wasserman, "Applied Linear Statistical Models", 1974, Irwin]. Since the sample size is balanced, the Tukey test can be used and is expected to be most sensitive.

In the third and final experiment, the 0-H option in the 0-N feature will be tested against 0-N feature only. As in the test for N, 6 subjects will be used for this analysis.

Appendix A.

Source	Sum of Square	d.o.f
Treatment		p-1
Problems		n(p-1)
Subjects		n(p-1)
Error		n^2p^2-p-2n(p-1)

Where,

p : Number of subjects, problems

n : Number of replications

ANOVA Table

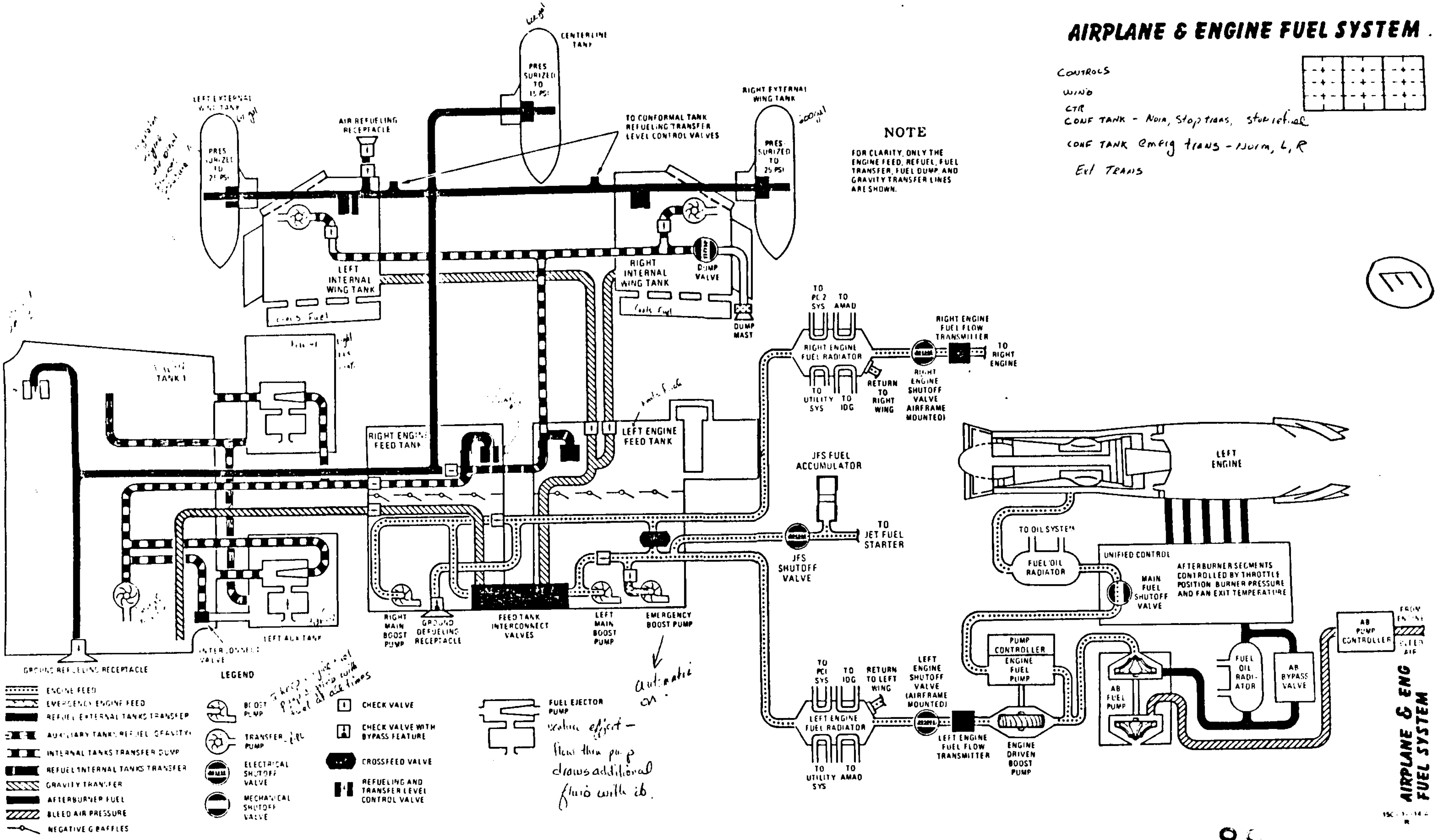
for Replicated Latin Square Design without Interaction

D

```
;fuel.lsp
; physical modeling representation and manipulation
; John M. Hammer
; 3/10/87
;(component
;  (name ())
;  (type ())
;  (ports
;    (sf-list
;      (port
;       (type ())
;       (name ())
;       (pressure ())
;       (flow ())
;       (connection
;         (tie-point
;           (component-name ())
;           (port-name ())
;           )
;         )
;      (port
;       (type ())
;       (name ())
;       (pressure ())
;       (flow ())
;       (connection
;         (tie-point
;           (component-name ())
;           (port-name ())
;           )
;         )
;       );sf-list
;  (state-variables
;    (sf-list
;      (state-variable
;       (mass ())
;       )
;      )
;    )
;  (parameters
;    (sf-list
;      (parameter
;       (resistance ())
;       (value ())
;       )
;      (parameter
;       (volume ())
;       (value ())
;       )
;  (behaviors
;    (sf-list
;      (behavior
;       (cond (<expr>))
;       (eqns (sf-list <ar> <ar> <ar>))
;       )
;      (behavior
;       (cond ((<expr>))
;       (eqns (sf-list <ar> <ar> <ar>))
;       )
;      );sf-list
;
; <expr>
; an expr is either an ar (defined below) or the and of
; a list of ars:
;       <expr>  ::= <ar>
;               ::= pand <ar> <ar> ... <ar>
; <ar>
; an algebraic relationship, which could be an equation or an
; inequality (possibly a constraint)
; examples:
;       a = 1           (peq a 1)
;       b < 3           (p< b 3)
;       x+y=z-q         (peq (p+ x y) (p- z q))
;
; in ports and parameters, there are slot names that are physical
; dimensions (e.g., resistance, pressure, flow)
; an example of a valve
;(component
;  (name (valve14))
;  (type (valve))
;  (ports
;    (sf-list
;      (port
;       (type (liquid))
;       (name (in-port))
;       (pressure (in-pressure))
;       (flow (flow))
;       (connection
;         (tie-point
;           (component-name (pipe-7))
;           (port-name (left-port))
;           )
;         )
;      (port
;       (type (liquid))
;       (name (out-port))
;       (pressure (out-pressure))
;       (flow (flow))
;       (connection
;         (tie-point
;           (component-name (pipe-4))
;           (port-name (right-port))
;           )
;         )
;       (port
;         (type (electrical))
;         (name (control))
;         (voltage (v-in))
;         (connection
;           (tie-point
;             (component-name (wire-3))
;             (port-name (left-end))
;             )
;           )
;         )
;       );sf-list
;  (parameters
;    )
;  (behaviors
;    (sf-list
;      (behavior
;       (cond ((peq v-in 'high)))
;       (eqns
;         (sf-list
;           (peq in-pressure out-pressure)
;           )
;         )
;       )
;      (behavior
;       (cond ((pneq v-in 'high)))
;       (eqns
;         (sf-list
;           (peq flow 0)
;           )
;         )
;       )
;      );sf-list
;    );behaviors
;  );component
; example of a tank
;(component
;  (name (tank13))
;  (type (tank))
;  (ports
;    (sf-list
;      (port
;       (type (liquid))
;       (name (in-port))
;       (pressure (in-pressure))
;       (flow (in-flow))
;       (connection
;         (tie-point
;           (component-name ())
;           (port-name ())
;           )
;         )
;      (port
;       (type (liquid))
;       (name (out-port))
;       (pressure (out-pressure))
;       (flow (out-flow))
;       (connection
;         (tie-point
;           (component-name ())
;           (port-name ())
;           )
;         )
;       );sf-list
;  (parameters
;    (sf-list
;      (parameter
;       (mass (maximum-mass))
;       (value (1700))
;       )
;  (state-variables
;    (sf-list
;      (state-variable
;       (mass (contents))
;       )
;      )
;    )
;  (behaviors
;    (sf-list
;      (behavior
;       (cond
;         (pand
;           (p< contents maximum-mass)
;           (p< 0 contents)
;           )
;         )
;       (eqns
;         (sf-list
;           (peq (pd/dt contents) (p- inflow out-flow))
;           (peq in-pressure (p* contents .31))
;           (peq out-pressure (p* contents .31))
;           )
;         )
;       )
;      (behavior
;       (cond
;         (peq contents maximum-mass)
;         )
;       (eqns
;         (peq in-flow out-flow)
;         (peq in-pressure out-pressure)
;         )
;       )
;      (behavior
;       (cond
;         (peq contents 0)
;         )
;       (eqns
;         (peq out-flow 0)
;         )
;       )
;      );sf-list
;    );behaviors
;  );component
;
;syntax of slot-filler objects
; <sf>          ::= ( <header> ( <slot> ( <filler> )) ( <slot> ( <filler> )) ...
; <header>      ::= <symbol>
; <slot>        ::= <symbol>
; <filler>      ::= <atom>
;               ::= <sf>
;               ::= sf-list <sf> <sf> ... <sf>
;               ::= an expression to be eval-d (this is kludgy)
; <symbol>      ::= denotes a lisp symbol
;
;
;
```

AIRPLANE & ENGINE FUEL SYSTEM

CONTROLS
WING
CTR
CONF TANK - Norm, Stop trans, stop refuel
CONF TANK emerg trans - Norm, L, R
Ext TRANS



NOTE

FOR CLARITY, ONLY THE ENGINE FEED, REFUEL, FUEL TRANSFER, FUEL DUMP, AND GRAVITY TRANSFER LINES ARE SHOWN.

Figure FO-4

F

A DEEP-REASONING AID FOR DEEP-REASONING FAULT DIAGNOSIS

Wan C. Yoon and John M. Hammer
Center for Man-Machine Systems Research, Georgia Institute of Technology,
Atlanta, Georgia 30332

ABSTRACT

• Wan C. Yoon and John M. Hammer, 1987. A deep reasoning aid for deep-reasoning fault diagnosis. Human-Computer Interaction, Vol 2 (G. Salvendy, ed.)

The design and an experimental evaluation are presented for an intelligent aid for a human operator who must diagnose a novel fault in a physical system. A novel failure is defined as one that the operator has not experienced in either real system operation or training. When the operator must diagnose a novel fault, deep reasoning about the behavior of the system is required. The aid contains features that support such reasoning. One of these is a qualitative, component-level model of the physical system. Both the aid and the human are able to reason causally about the system in a cooperative search for a diagnosis. The human diagnostic performance improved by almost a factor of two when the aid presented the information of observed system behavior or the difference between observed and normal behavior.

1 INTRODUCTION

In highly automated systems, the human operator is primarily a monitor and supervisor [Rasmussen, 1983]. An important monitoring function is diagnosing equipment faults, a difficult task in automated systems. The current approach to fault diagnosis is to train the operator to deal with relatively common faults. The training might teach the operator to use symptoms to distinguish faults and to follow procedures to correct them. While this approach should be successful with common faults, it does not support diagnosis of novel faults. A common sense but unsuccessful approach to help operators diagnose novel fault is to teach them the principles of operation of the system. With this theoretical knowledge, the operators should be able, in principle, to diagnose any failure. Unfortunately, there is little evidence that theoretical knowledge helps operators diagnose failures [Morris and Rouse, 1985a,b]. A logical consequence of this observation might be to put theoretical knowledge into the aid rather than the operator.

Our aid is based on deep, causal reasoning about the system. There are several advantages to this approach. First, novel fault diagnosis is normally considered to be knowledge-based reasoning [Rasmussen, 1983]. Hence, it seems appropriate for an intelligent aid to reason causally. Second, this approach should be more reliable and robust. The system knowledge is represented at the component level. Because components are small and comprehensible, it should be possible to create representations that are correct, perhaps even provably so. These points support the belief that causal reasoning can cover a wider range of faults [Davis, 1984].

In spite of the power of the intelligent aid, we believe there are several reasons to keep the human in command of the problem solving. First, diagnosing a novel failure may require the human to extend the aid's model. Second, when diagnosis involves operating the system (e.g., opening valves, starting motors), it would be better to leave these operations to the human. Third, causal reasoning is slow because the diagnosis problem is a combinatorial search. It may be that the human and the aid may be better able to find a solution cooperatively than either can alone. This is possible, even necessary, for two reasons. The human has better pattern recognition capabilities and can make inductive leaps. Second, the human may need to resolve ambiguities inherent in the aid's model.

In the subsequent sections of this article, we will discuss the system and the experimental task, the interface, the model of human information processing, the aids, and the experimental results.

2 THE SYSTEM AND THE TASK

2.1 The System

The Orbital Refueling System (ORS), a NASA-designed payload on the Space Shuttle, was selected for study [NASA, 1985]. The function of the ORS is to refuel orbiting satellites with hydrazine, with the objective of extending their useful service life. As shown in Figure 1, the ORS fluid system contains a variety of components such as tanks, valves, pipes, etc. The operator controls the simulated ORS by opening and closing valves. Transferring fuel from propellant tank 1 to propellant tank 2 might proceed as follows. First, tank 2 pressure is reduced by momentarily opening valves 10, 11, 13, and 17. Second, tank 1 is pressurized by opening valves 1, 3, and 7. Gaseous nitrogen will flow out of the two small supply tanks, be pressure regulated, and fill tank 1 on one side of the bladder. To transfer fuel to tank 2, valves 5, 14, 15, 16, and 9 would be opened. Because this version of the ORS was for demonstration purposes, all transfers take place between the two large tanks rather than to a satellite fuel tank. There are several assemblies whose purpose was not explained in the above example. The relief valves RV1 and RV2 serve as a safety pressure relief. Check valve CV1 prevents backflow into the gas system. The bladders in tank 1 and 2 serve to isolate the fuel from the propellant and also to contain the fuel in the weightlessness of space. Some components (e.g., valves 10 and 11) may seem redundant; they are so by design for two failure tolerance.

2.2 The Diagnosis Task

The operator's task is to diagnose the failure in the system. This requires the operator to manipulate and observe the system, because a diagnosis cannot be determined uniquely from an observation of a state vector at a single point in time. The diagnosis task is difficult for the following reasons. First, all component testing must be done in the context of the system. It is not possible to remove a component for isolated testing. Thus, every diagnostic test requires nontrivial interpretation. Second, the data are limited and may contain one or more errors. There are seven pressure sensor readings and fourteen commanded valve positions. Both can contain an error. A pressure sensor may report a false reading or a valve may disobey its command. The consequences are that an unaided diagnosis can easily require ten minutes.

3 AIDING WITH A QUALITATIVE MODEL

This section describes the interface, our model of operator information processing, and the aids. The interface has four windows: schematic, interac-

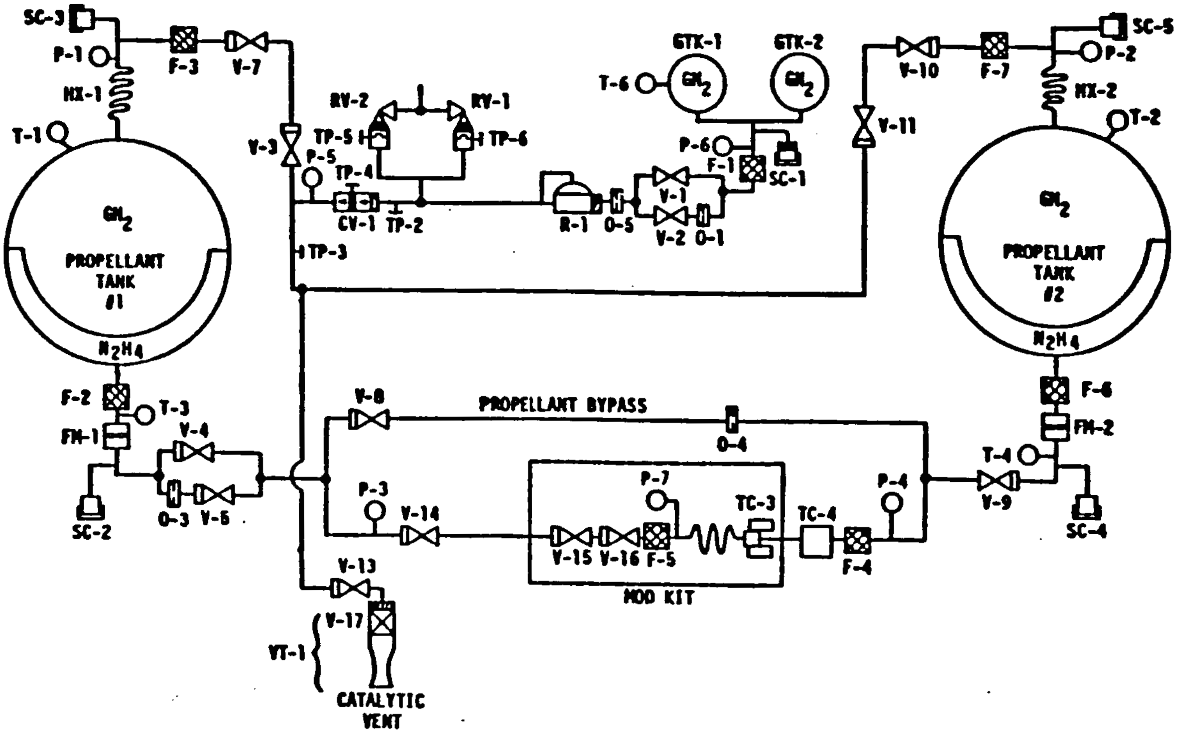


Figure 1. The Orbital Refueling System (ORS).

tion, sensor display, and hypotheses.

The schematic window displays a schematic diagram of the ORS. The schematic always shows the commanded states of the valves. Certain forms of aiding (described below) change the display of paths along which mass may flow. The appearance of the schematic is similar to Figure 1.

The interaction window is where the operator's commands appear. The commands available to the operator include the following:

1. Opening and closing valves.
2. Comparing two pressures. On a real physical system, the numerical pressure could be displayed on the schematic. When a qualitative model is used, there is no scale in general to which a pressure can be referred. Instead, the subject may request the relationship (<, =, >) between two pressures or between a pressure and a nominal reference pressure such as absolute zero or the regulator's design set point.
3. Display of the first derivative of pressure (positive, zero, or negative).
4. Turning the what-if model aid (described below) on and off, and stating hypotheses to the what-if model aid. When the what-if model is on, the open, close, and comparison commands apply both to the system and the what-if model.

The sensor display contains the output from the comparison command: the relationship between two pressures or the first derivative of a pressure. The what-if model, if activated, has its corresponding output displayed side-by-side with the system model.

The hypotheses window will display any hypotheses that the operator expresses through commands in the interaction window.

4 A MODEL OF OPERATOR INFORMATION PROCESSING

4.1 Observation of Strategies

Our model of operator information processing directly influenced the design of the aids. From the observation of diagnostic behavior, we had identified three strategies that subjects used: hypothesis-driven evaluation, data-driven evaluation, and topographic search [Yoon and Hammer, 1987]. Hypothesis-driven evaluation starts with the planning of a test procedure for a given hypothesis. A test plan would be diagnostic if, given that the hypothesis is true, the response of the system to the test is unique to the hypothesis. When a sufficiently diagnostic test has been planned, the test is executed and its result evaluated. Because the hypothesis needs to be explicit enough to enable the prediction of its resulting system behavior, this strategy is mostly used in the later phase of diagnosis.

With data-driven evaluation, the subject first examines a piece of data to determine if it is worth closer attention. This examination is done by comparing the data to the expected system behavior. If the data turns out to be unexpected (i.e., not explained in terms of previously observed symptoms or normal behavior), then hypotheses are formulated to explain the data. Whether the formulation is successful, this piece of data is remembered as another symptom to be used later during diagnosis. Since this strategy does not require a well-formed hypothesis, it was heavily employed in the initial phase of diagnosis.

Topographic search follows the connections between components to track down the source of the malfunction. In contrast to hypothesis-driven and data-driven evaluation, it does not appear to require as deep a reasoning about device behavior. Thus, it is easier.

4.2 Types of information processing

As frequent parts of some of the above strategies, the operator needs presumably to do the following types of information processing. First, the operator must envision the normal behavior (i.e. no failures) of the system. Second, the operator uses external, observable information (i.e., pressure information) to determine unobservable, internal behavior (i.e., presence of a mass flow, a leak somewhere in a path). Third, the operator must form the difference between the observed and normal system behavior. These three forms of processing could be termed N (normal), O (observed), and O-N (observed minus normal).

The aids parallel the above three forms of processing. N and O aiding are intended to help the operator with N and O processing, respectively. Both are displayed in the same way. The schematic display is modified to show both mass flow paths (the movement of either gas or liquid) and equal pressure paths. The determination of these paths is from a system model (N) or pressure observations (O) available to the aid. The aid has exactly the same information as does the operator.

O-N aiding is the difference between observed and normal behavior. This information is displayed in the sensor display window in the form of suggested data observation commands. This form of aiding was also predicted to be useful based on earlier observations [Yoon and Hammer, 1987]. Subjects appeared to have difficulty selecting effective data to observe.

A fourth form of aiding, O-H, is closely related to the third, O-N. O-H (observed minus hypothesized) aiding displays the difference (as described above) between the observed behavior and a system with one or more hypothetical failures. This aid allows the operator to set a hypothesis. If the hypothesis is correct, there will be no difference between observed and hypothesized behavior. This aid gives the operator an unambiguous interpretation of the correctness of a hypothesis. It does not, however, tell the operator how to modify the hypothesis if it is incorrect.

5 EXPERIMENTS AND RESULTS

5.1 Procedure

Two experiments were conducted to evaluate the aids. A comparison of N versus unaided performance was first tested since we had earlier observed that most subjects found it confusing or irrelevant. The more prospective aids, O and O-N, were evaluated in the second experiment. Six and nine engineering students participated in the first and the second experiment, respectively.

Two training sessions preceded the experimental session. The first session was self-paced instruction on basic fluid dynamics and the operation of the ORS. In the second session, the subjects practised testing various hypotheses and solved five diagnostic problems both with and without the aids. The purpose of these experiments intentionally limited the useful range of diagnostic skill of subjects. An overtrained subject tends to develop some mechanistic diagnosis procedures. These may replace the deep reasoning about the system and deal with the problems as routine failures rather than novel ones. With too little training, the subject's performance would reflect more of deficiency in knowledge than the difficulty of the problem solving. For these reasons, the experimenter interacted with the subjects in both training sessions to insure proper understanding of the material.

The subjects started the experimental session with several additional, warm-up exercises and solved six main problems. Keystrokes and verbal protocols were collected. The performance measures were the time to diagnose (TTD) and the number of information gathering actions (#IGA). Problem and subject were blocking variables. Each subject solved the problems with an equal number of different aiding levels. A replicated Latin square was used. Order effects were counterbalanced. Three subjects formed a group, which received the same order of aids, to serve as replications for the evaluation of interaction terms in both experiments [Winer, 1962, pp. 538-543].

5.2 Results

The results of significance tests were same with TTD and #IGA. The effect of N aiding was somewhat negative, though not significant. Most subjects said after their sessions that the aid N was rather confusing or that it was not the information they were seeking during the diagnosis. Both O and O-N aids showed a positive improvement in diagnostic performance at the 0.05 significance level. The effects of both blocking variables, subject and problem, were significant. But, there was no significant interaction between any two variables. Residual analysis revealed that logarithmically transformed data better satisfied the homogeneous variance assumption. No test results, however, were changed by the transformation. It was shown that O-N and O shortened TTD on the average by 42% and 34%, respectively. The aiding effects appeared similar in #IGA: 44% decrease with O-N, 40% with O.

5.3 Additional observations

The following observations, while not the result of hypothesis testing, were also made during the course of the experiment. The aids more benefited the problem solving earlier in the diagnosis. This was expected because one of the effects of O and O-N was to reveal abnormal system responses, and thus to stimulate the subject to launch a data-driven evaluation. In fact, O-N aiding obviously encourages the subject to select meaningful data,. Toward the end of diagnosis, the subjects developed explicit hypotheses (i.e. hypothesis-driven evaluation), and tended to be too heavily involved in their own testing procedure to pay attention to the aid. In fact, the aiding information is usually no longer relevant to the subjects´ highly detailed hypothesis testing. To aid the final phase of diagnosis, the aid needs to know the operator´s hypothesis. Then, the aid could run a modified qualitative model according to the hypothesis (H) and calculate its deviation from the observed behavior (O). The difference O-H may be more relevant than O-N to the later phase of diagnosis.

6 CONCLUSION

An aiding approach has been described and evaluated for novel fault diagnosis in complex systems. To the best of our knowledge, this approach is unique in the following ways. First, the emphasis is on novel rather than routine faults. Second, it contains a qualitative model that may correspond to the human´s internal model of the system. This model represents knowledge only of how the system behaves. Therefore, this aiding approach does not rely on proceduralized knowledge. Third, the qualitative model is the basis for much of the aiding that takes place.

The qualitative model was used to help different tasks of human information processing. Presentation of observed system behavior (O) improved the diagnostic performance of subjects, while that of normal system behavior (N) does not. One implication is that the prediction of current actual system behavior is a task that needs more help. Aiding of envisioning normal system behavior according to commanded physical configuration is less effective and, when emphasized saliently, seems to interfere with the diagnostic reasoning. Pointing out the abnormality in the observed system behavior (O-N) was at least as effective as O.

More generally, the experiment confirmed that a deep reasoning diagnosis can be aided, without disturbing the human diagnostic procedure, by providing relevant information. It should be emphasized that this was possible through an understanding of the operator's information needs and that a qualitative model could be used to generate the information that seemed to be well accepted for augmenting the human's mental model.

7 ACKNOWLEDGMENT

This research was supported by NASA-Ames Research Grant NAG 2-123. Dr. Everett Palmer is the technical monitor.

8 REFERENCES

Davis, R., 1984. Diagnostic reasoning based on structure and behavior. Art. Intel., 24: 347-410.

Morris, N.M. and Rouse, W.B., 1985. The effects of type of knowledge upon human problem solving in a process control task. IEEE Trans. Sys., Man. and Cyber., 15(6): 698-707.

Morris, N.M. and Rouse, W.B., 1985. Review and evaluation of empirical research in troubleshooting. Human Factors, 27(5): 503-530.

NASA Johnson Space Center, 1985. Orbital refueling demonstration system description. Program Development Office.

Rasmussen, J., 1983. Skills, rules, and knowledge: signals, signs, and symbols, and other distinctions in human performance models. IEEE Trans. Sys., Man. and Cyber., 13-3: 257-266.

Winer, B.J., 1962. Statistical Principles in Experimental Design. McGraw-Hill, New York/San Francisco/Toronto/London.

Yoon, W.C. and Hammer, J.M., 1987. Using a Qualitative Model to Aid the Operator. to appear in the IEEE Trans. Sys., Man. and Cyber. during Novel Fault Diagnosis. Proceedings of the International Conference on Cybernetics and Society, pp. 1222-1227.