DOCUMENT RESUME

ED 296 716 IR 013 399

AUTHOR	Towne, Douglas M.; And Others
TITLE	Intelligent Maintenance Training Technology. ONR Final Report. Technical Report No. 110.
INSTITUTION	University of Southern California, Los Angeles. Behavioral Technology Labs.
SPONS AGENCY	Navy Personnel Research and Development Center, San Diego, Calif.; Office of Naval Research, Arlington, Va.
PUB DATE	Mar 88
CONTRACT	N00014-85-C-0040
NOTE	65p.
PUB TYPE	Reports - Evaluative/Feasibility (142)
EDRS PRICE	MF01/PC03 Plus Postage.
DESCRIPTORS	*Computer Assisted Instruction; *Computer Simulation; Computer Software; Equipment Maintenance; *Expert Systems; *Man Machine Systems; *Models; Postsecondary Education; Problem Solving; *Technical Education

ABSTRACT

This report describes the Intelligent Maintenance Training System (IMTS), a set of software tools for composing and delivering simulation-based technical training. The goal in developing IMTS was to generate instructional interactions from device models composed of instances of generic objects. Problem selection in IMTS relies on the use of a normative model that reflects the structure of the target device. Proficiency measures are maintained for each student in terms of this model and support the selection of problems of appropriate difficulty and type. IMTS incorporates a formulation of generic diagnostic expertise, termed Profile, that: (1) explains the significance of particular test outcomes and remediates student beliefs about symptom implications; (2) recommends what to do next, taking completed tests into account; (3) assesses student proficiency; and (4) debriefs the students following each fault isolation exercise. A number of findings and recommendations concerning alternative approaches to simulation, student monitoring, problem selection, and student aiding are presented. The next steps for IMTS development are also outlined, including techniques for simulation devices whose complete functional behaviors cannot be reasonably specified, and the addition of features to support procedural training. The text is supplemented by 23 figures. (24 references) (Author/EW)

IR

March 1988

Technical Report No. 110

ONR Final Report:

Intelligent Maintenance Training Technology

Douglas M. Towne
Allen Munro
Quentin A. Pizzini
David S. Surmon
James Wogulis

Behavioral Technology Laboratories
Department of Psychology
University of Southern California

Sponsored by

Office of Naval Research
Cogitive Science Research Programs

and

Navy Personnel Research and Development Center
Under Contract No. N00014-85-C-0040

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED
Reproduction in whole or in part is permitted
for any purpose of the United States Government

ONR Final Report:

Intelligent Maintenance Training Technology

Douglas M. Towne
Allen Munro
Quentin A. Pizzini
David S. Surmon
James Wogulis

March 1988

Technical Report No. 110

Behavioral Technology Laboratories
University of Southern California

Sponsored by
Office of Naval Research
and
Navy Personnel Research and Development Center
Under Contract No. N00014-85-C-0040

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
"Approved for Public Release:
Distribution Ullimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
Technical Report No. 110

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
Behavioral Technology
Laboratories - USC

6b. OFFICE SYMBOL (If applicable)

7a. NAME OF MONITORING ORGANIZATION
Cognitive Science Research Programs
Office of Naval Research (Code 1142cs)

6c. ADDRESS (City, State, and ZIP Code)
1845 S. Elena Ave., 4th flr.
Redondo Beach, CA 90277

7b. ADDRESS (City, State, and ZIP Code)
800 N. Quincy Street
Arlington, VA 22217-5000

8a. NAME OF FUNDING / SPONSORING ORGANIZATION
Navy Personnel Research and
Development

8b. OFFICE SYMBOL (If applicable)

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
N00014-85-C-0040

8c. ADDRESS (City, State, and ZIP Code)
Code P-306
San Diego, CA 92152

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
61153N	RR04205	RR04206-01	NR-535-001

11. TITLE (Include Security Classification)
ONR Final Report: Intelligent Maintenance Training Technology

12. PERSONAL AUTHOR(S)
Douglas M. Towne, Allen Munro, Quentin A. Pizzini, David S. Surmon, James L. Wogulis

13a. TYPE OF REPORT
Final

13b. TIME COVERED
FROM 2-87 TO 3-88

14. DATE OF REPORT (Year, Month, Day)
88/03/31

15. PAGE COUNT
32

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
05	09	08

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Artificial Intelligence, Graphical Simulation, Troubleshooting Expertise, Simulation Training, Representing Device Behavior

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The Intelligent Maintenance Training System (IMTS) is a set of software tools for composing and delivering simulation-based technical training. The goals in developing IMTS was to generate instructional interactions from device models composed of instances of generic objects.

Problem selection in IMTS relies on the use of a normative model that reflects the structure of the target device. Proficiency measures are maintained for each student in terms of this model and support the selection of problems of appropriate difficulty and type.

IMTS incorporates a formulation of generic diagnostic expertise, termed Profile, that 1) explains the significance of particular test outcomes and remediates student beliefs about symptom implications, 2) recommends what to do next, taking completed tests into account, 3) assesses student proficiency, and 4) debriefs the student

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL
Dr. Susan Chipman

22b. TELEPHONE (Include Area Code)
(202 696-4318

22c. OFFICE SYMBOL
ONR 1142cs

m 1473, JUN 86 Previous editions are obsolete.

Abstract Continued

following each fault isolation exercise.

A number of findings and recommendations are listed. The next steps for IMTS development are also outlined, including techniques for simulation devices whose complete functional behaviors cannot reasonable be specified and the addition of features to support procedural training.

ABSTRACT

The Intelligent Maintenance Training System (IMTS) is a set of software tools for composing and delivering simulation-based technical training. The goal in developing IMTS was to generate instructional interactions from device models composed of instances of generic objects.

Problem selection in IMTS relies on the use of a normative model that reflects the structure of the target device. Proficiency measures are maintained for each student in terms of this model and support the selection of problems of appropriate difficulty and type.

IMTS incorporates a formulation of generic diagnostic expertise, termed Profile, that 1) explains the significance of particular test outcomes and remediates student beliefs about symptom implications, 2) recommends what to do next, taking completed tests into account, 3) assesses student proficiency, and 4) debriefs the student following each fault isolation exercise.

A number of findings and recommendations are listed. The next steps for IMTS development are also outlined, including techniques for simulating devices whose complete functional behaviors cannot reasonably be specified and the addition of features to support procedural training.

TABLE OF CONTENTS

7

LIST OF FIGURES

ACKNOWLEDGEMENTS

Development of the Intelligent Maintenance Training System was supported by the Cognitive Science Research Programs of the Office of Naval Research and by the Navy Personnel Research and Development Center. We thank Susan Chipman and Henry Halff of ONR and Vern Malec of NPRDC for their support of this work.

The Profile model of expert diagnostic behavior was originally developed under funding from the Engineering Psychology Group of the Office of Naval Research. We thank Martin Tolcott and Gerald Malecki for their support of this work.

William Johnson of Search Technology and Ronald Renfro of ManTech Mathetics Corporation have provided subject-matter expertise for the first applications of the IMTS. Johnson has also participated in the preliminary evaluations of the simulation authoring facilities and has contributed to the student interface design.

Portions of IMTS were programmed by Norman Chen of the University of Southern California.

SECTION I. INTRODUCTION

This is the final technical report covering development of the Intelligent Maintenance Training System (IMTS) developed at Behavioral Technology Laboratories over the past three years.

The IMTS serves three purposes. First, it is a demonstration of the feasibility of generating device-specific maintenance instruction from generic troubleshooting intelligence applied to device specifications. Second, it provides a tool for the development and delivery of maintenance training for such applications as the SH-3H Helicopter Bladefold system and a WSC-3 Satellite Communications System. Third, it provides an extendable environment for further research on a variety of topics, including the application of direct manipulation methodologies to graphic simulation composition; the effectiveness of generated instruction as opposed to authored instruction; and appropriate roles for interactive graphics in technical training.

The Goal: Maintenance Instruction Derived from Device Models

An overriding goal in the development of IMTS has been to generate device-specific training using device-general intelligence applied to technical specifications of a particular device. Of course, hand tailored instructional materials are valuable and even necessary in many contexts. Still, there are considerable advantages to generated instruction. Where instruction can be effectively generated, it may be possible to guarantee greater thoroughness, completeness, and consistent accuracy than is possible with purely human-generated instructional materials. Generated instruction allows meaningful interactions when a huge number of situations and requirements could be encountered and allows production of training courseware at a reduced cost, in comparison to manually authored instruction.

Creating instructional interactions at run time based on a device description has a number of advantages, both for the student and for those who must develop, maintain, and administer the training system. Chief among the advantages for the student is that a wide range of interactions are possible. This makes it feasible for students to control many of the surface characteristics of their interactions with the

IMTS tutor/advisor. Students can utilize options of a data-driven system that would not be feasible in a system in which all the interactions are pre-authored. For example, IMTS students can insert simulated malfunctions for which the human author has prepared no instructional materials and then experiment with the simulated device exhibiting that failure mode.

For the developers and maintainers of a technical training package, there are also great advantages to basing training on a device model. Many different kinds of aiding and instructional interchanges can be delivered to the student without having to be separately authored. If the target device is modified, the corresponding change can be made to the description, and all the instructional elements related to the change will automatically have the appropriate form. Similarly, if an error in the expert author's understanding of the device description is detected, it can usually be repaired with a simple change to the technical specification.

Our objective was to make this process as easy as possible. The IMTS development system consists of a number of generalized tools for describing devices (a "device", as used here, is any collection of component parts). Graphics editors are used to draw the components of a device and position them appropriately in scenes. A behavior editor is used to describe the ways in which generic objects function in normal and failed states. Most of the actual troubleshooting advice and instruction provided to the student is created at run time by an IMTS generic expert module that has access to the device description and to the student's diagnostic performance.

Approach

One key to the generation of instruction in the IMTS is the separation of the generic intelligence required to instruct corrective maintenance from the device-specific technical information (knowledge) that characterizes each particular system. The generic intelligence built into the IMTS consists of 1) diagnostic expertise, and 2) instructional expertise.

Because the instructional intelligence is provided in the IMTS, it is possible for technical experts to supply the necessary specifications to support training without also requiring instructional expertise, diagnostic expertise, or programming skills. The drawback of the approach is that users cannot alter the instructional strategy or the diagnostic approach, if they feel that these could be improved or adapted to better meet particular requirements. A future version of IMTS may incorporate tools for customizing the instructional approach and the diagnostic strategy.

We now briefly describe the instructional approach and diagnostic strategy employed in the IMTS.

The Instructional Orientation

The IMTS was designed to provide troubleshooting practice to technicians already trained in general principles (of electronics or hydraulics) and already familiar with the appearance, organization, and function of the device they are learning to maintain.

Instruction is simulation-oriented, i.e., the student learns by attempting to perform troubleshooting tasks on a simulation of the target device. For each exercise, the student is presented with a failure report and a simulation of the device, containing a failure selected by the training system. The student performs tests and replacements on the simulation until he or she claims that the system has been restored to normal operation. When necessary, or upon student request, advice and commentary about the student's troubleshooting actions are provided.

The IMTS training philosophy is built upon a foundation of research in intelligent tutoring. Twenty-one principles from this research guided the development of instructional decisions within the training context. In brief, these are the principles, listed with the research studies that justify them:

1. Instruction should be relevant to the problem-solving context.
   (Tulving, 1983; Tulving & Thompson, 1973)

2. Provide immediate feedback on errors.
   (Bilodeau, 1969; Skinner, 1958)

3. Sustain the simulation session by postponing instructional content which would obviate the motivation for the session. (Don't give it all away!)

4. Provide quick response.

5. Explicitly identify the goal structure of the problem domain. (Anderson, Farrell, & Sauers, 1984)

6. Minimize the student's working memory load. (McKendree, Reiser, & Anderson 1984)

7. Prevent superstitious behavior.

8. Help students to approach target skills by successive approximations. (Anderson, 1983; Anderson, Farrell, & Sauers, 1984)

9. Protect students from building extended chains of misconceptions.

10. Protect students from negative consequences of appropriate actions.

11. Instruct opportunistically.

12. Maintain the credibility of the tutor.

The remaining principles are taken from (Burton & Brown, 1982).

13. Before giving advice, be sure that the student needs it.

14. When illustrating an issue, use an example for which the correct move is dramatically superior to the move made by the student.

15. After remediating, give the student an opportunity to exercise the new knowledge.

16. If the student is about to fail, interrupt and provide moves that will prevent failure.

17. Never intervene and tutor on two consecutive student moves.

18. Don't prevent student discovery by overtutoring.

19. Intervene on exceptional successes, as well as on failures.

20. Always have the artificial expert play an optimal game.

21. Provide help in several layers.

The largely non-invasive, non-controlling nature of the IMTS instructional system can be attributed to adherence to many of these principles. A student who has been doing fairly well in a problem will occasionally perform a rather poor test. If progress has been generally good in comparison with expert performance, the IMTS will remain silent and let the student continue. In this respect, IMTS behaves much like many human tutors, who often allow students to continue when their behavior is not optimal, so long as progress is being made. This hands-off approach gives students an opportunity to practice without constant interruption. Naturally, performance is still monitored in this silent mode, and aid is given to help students avoid long unproductive periods. Further discussion of these principles can be found in an earlier report (Towne, Munro, Pizzini, Surmon, and Johnson, 1985).

Profile — a Generic Model of Diagnostic Expertise

Under funding from the Engineering Psychology program of the Office of Naval Research, BTL developed a generic expert diagnostician called Profile (Towne, Fehling, & Bond, 1981; Towne, 1984; 1985; 1986) and conducted empirical studies of its troubleshooting behavior in comparison with that of human experts (Towne, Johnson, & Corwin, 1982; 1983). The Profile model of diagnostic expertise is the foundation for a number of different research efforts at Behavioral Technology Laboratories (Figure 1).

Profile can be applied to generated fault data in several different ways. For example, it can be used in conjunction with a commercial computer-aided engineering (CAE) package to evaluate designs for their maintainability (Towne & Johnson, 1984; 1987). In the research reported on here, Profile is used to generate intelligent advice and commentary on troubleshooting actions made on a simulated device.

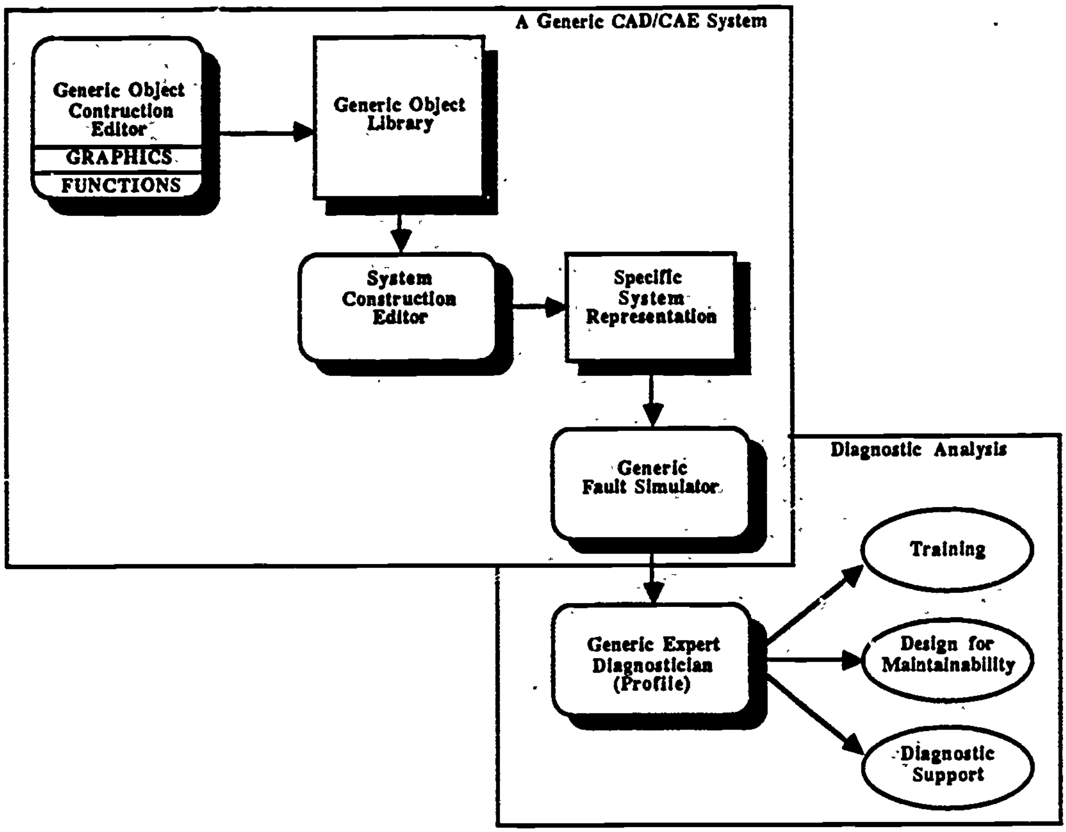


Figure 1. Profile Fills Multiple Roles

Training Configurations

The present IMTS has been implemented on Xerox Artificial Intelligence workstations, including the models 1108, 1186, and 1132. These computers are microcoded for Lisp, and they offer high speed graphical displays of moderately high resolution (at least 1024 x 800). All IMTS development/authoring tools, simulation routines, and instructional presentation routines were written in Interlisp-D, a Lisp dialect for which the Xerox machines have been optimized.

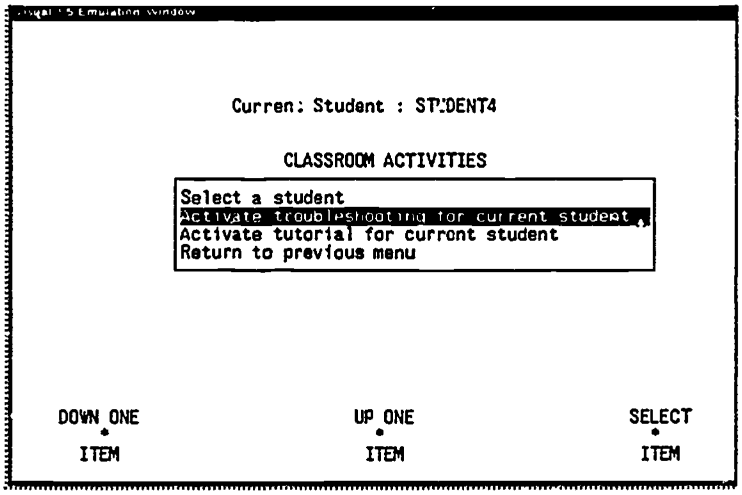
The IMTS is designed to work in two different training delivery environments, either 1) as an intelligent adjunct to some other training device, such as a high-fidelity simulator, or 2) as a stand-alone training system.

Adjunct Configuration

When IMTS is connected to an external simulator that adheres to the IMTS communications protocol, students can interact primarily with the external simulator, relying on the IMTS display for advice and ancillary instruction. The high-fidelity simulator involved in this dual-mode configuration might be a full-scale mock-up, a computer-controlled 2-D simulator, or just a videodisk unit.

At present, one external simulator, the Generalized Maintenance Trainer Simulator (GMTS) supports this communication protocol. GMTS is a microcomputer-based surface-behavior simulator that presents color images stored on videodisk under computer control(Towne, 1986; Munro, Towne, & King, 1980; Towne & Munro, 1981; Johnson, Munro, & Towne, 1981). Students use a touch-sensitive panel on the external monitor to change controls and to place test equipment probes on the displayed video images.

IMTS supports the textual communication features of GMTS by providing a window that acts as a virtual terminal display for the use of the GMTS software (Figure 2). This window provides a 25-line by 80-character screen that presents the text output of the GMTS computer, replacing the ASCII terminal that is used with GMTS simulations not augmented with IMTS instruction.



Current Student : STUDENT4

Figure 2. A Virtual Terminal Window Under the Control of GMTS

The GMTS simulator has no built-in instructional capacities when it is used in the simulated troubleshooting mode (it does provide a general-purpose instructional function as a separate facility). Communicating over an RS232 interface, the GMTS informs the IMTS of every action that the student takes. The IMTS recognizes what tests are being performed, and it evaluates them by calling Profile. Profile-based evaluations and Profile-generated suggestions provide the bulk of the instructional features available in the combined IMTS-GMTS mode.

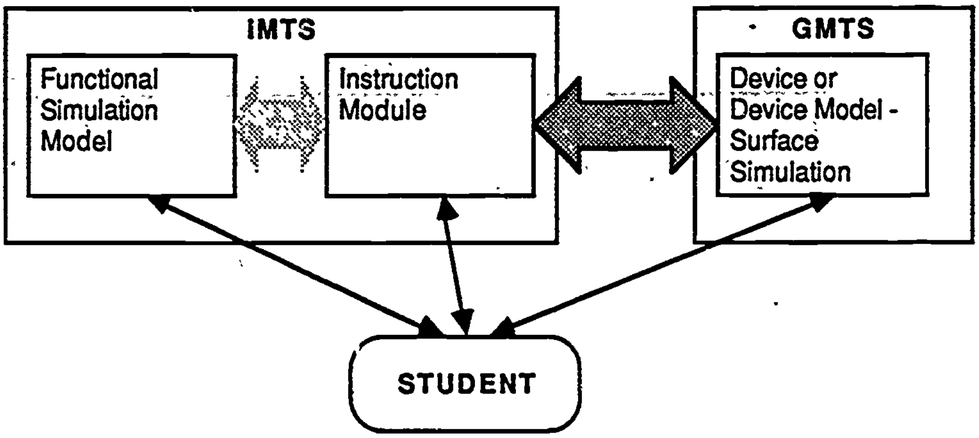


Figure 3. Student Interactions in the IMTS/GMTS Environment

Students can interact with at least three different modules in the integrated IMTS/GMTS environment (see Figure 3): 1) they can manipulate and observe the functional simulation shown on the IMTS graphics display, 2) they can interact with the instruction module in ways that will be described below, or 3) they can manipulate and observe the physical surface simulation portrayed by the GMTS using videodisk and graphics overlays.

In fact, the box labeled GMTS in the above figure is not constrained to be a videodisk-based simulator. It could as well be a flat panel or a 3-D simulator that has a computer interface that can communicate with IMTS using the IMTS/GMTS communications protocol. Even an actual device could conceivably be used, so long as it was modified so that it would report all user actions to the IMTS using the

IMTS communications protocol. Viewed this way, IMTS is an intelligent aiding station that can be used with any of a variety of training devices.

Stand-alone Configuration

In the stand-alone configuration, currently under development, the IMTS will perform all the simulation and instruction functions without a mediating GMTS computer system, as shown in Figure 4.

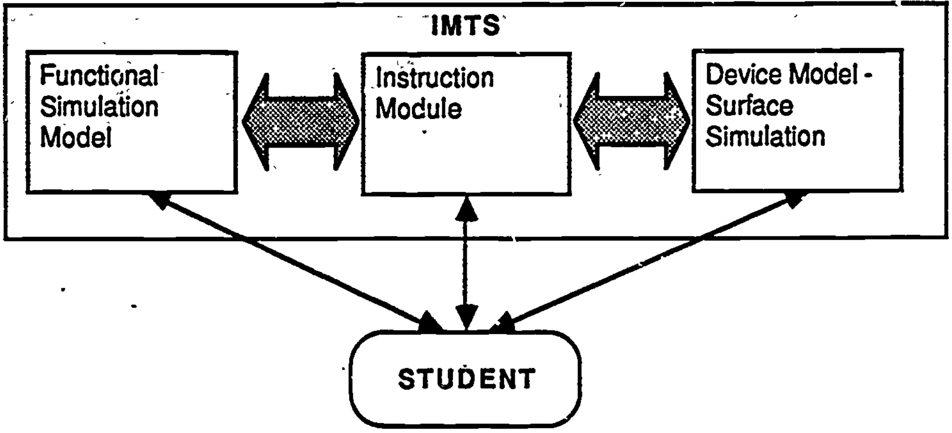


Figure 4. Student Interactions with Stand-alone IMTS

In this environment, the surface simulation currently provided by GMTS may be presented graphically on the IMTS computer display or may be displayed using a videodisk under the direct control of the IMTS computer.

IMTS Elements

Regardless of the configuration (adjunct or stand-alone), the primary elements in the IMTS design are these:

- a functional model of the device,
- graphical representations of the device, functional and/or physical,

- a model of the student,
- generic diagnostic expertise,
- generic instructional expertise.

Section II describes techniques for producing and using the functional device model, including graphical representations of the device; Section III describes the techniques for modeling the student and selecting appropriate problems; Section IV describes the IMTS processes for monitoring student proficiency and progress, and for providing appropriate assistance; and Section V presents conclusions.

SECTION II. MODELING AND REPRESENTING THE DEVICE

Virtually all of the IMTS instruction is derived from manipulations of a functional model. This model is produced automatically when the functional scenes are composed, i.e., underlying the graphical representation are all the generic object rules and connectivity data necessary to determine the behavior of the particular device. Interactions with the student may be based either on the functional representation or upon a physical representation that is created in terms of the functional form. We first discuss the functional model and its associated representation.

The Functional Model

The functional model is regarded as the primary medium for illustrating and explaining the behaviors of the target system. The model is represented to the student via computer graphic drawings that respond to the effects of the student's actions and to inserted failures. The first device we have modeled in this fashion is a complex helicopter bladefolding system in which the components are organized as thirteen interrelated schematic 'diagrams', or scenes. Figure 5 shows a portion of a representative scene from the Bladefold simulation set.

The student changes the setting of a control by selecting the desired new position with the mouse (all references to 'selecting' imply that the user positions the mouse cursor on the desired object or menu item and clicks the mouse button). The simulation is then updated, i.e., all effects of the switch change are determined, including effects that are off screen (on other scenes). This complete evaluation is necessary, since effects on the currently displayed scene may be caused by changes in portions of the simulation not in view (which were in turn caused by the switch change accomplished on screen). The currently displayed scene is then updated by displaying all new object states, including the switch that was changed. Because the non-dynamic background elements in the scene are not redisplayed, the user sees the changes in object states without visual disruption.

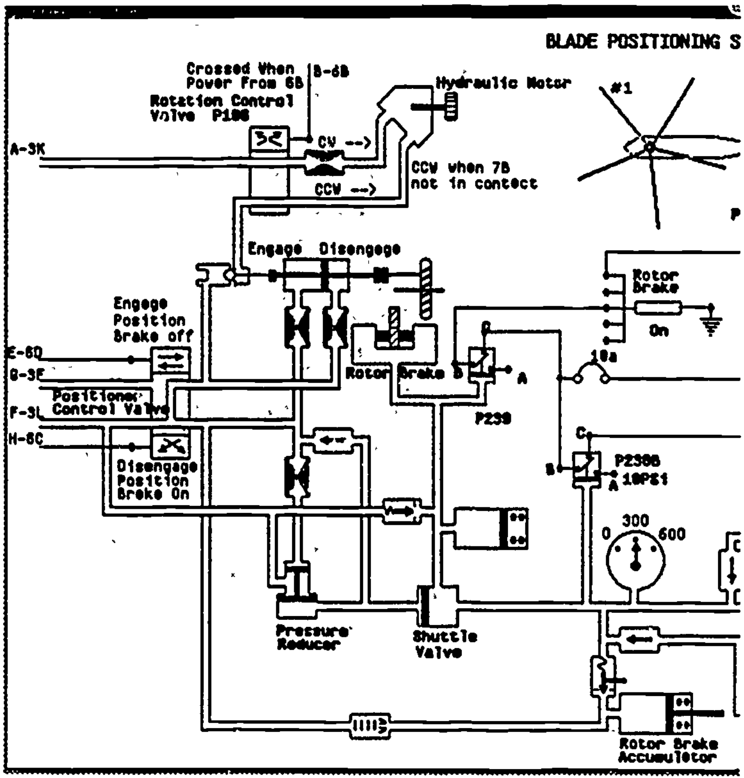


Figure 5. A Scene from the IMTS Bladefold Graphical Simulation

The total time to update the thirteen-scene Bladefold simulation, in response to a student action, is now approximately four seconds. This response time was attained only after extensive experimentation, analysis, and modification of alternative simulation algorithms.

The student can also observe the value at a test point by selecting the graphical object representing the test equipment to be used, and then selecting the test point, as displayed on the functional diagram. The test equipments currently provided are a multimeter, oscilloscope, and pressure gauge. Course developers can add new test equipments by defining them graphically and functionally, using the standard IMTS object authoring system, described below.

The simulation can represent a normally operating system, or one with one or more failures present. If there are failures present, they can either be known to

21

the student or they may be of unknown character, to be determined by the student's diagnostic actions. Students can replace simulated objects with ones known to be good. Upon replacing the malfunctioning component, the student will observe entirely normal test results (if there was only one failure present).

Students move through the scenes of a complex simulation by using a hierarchical map of the functional subsystems of the device. The map (Figure 6) is always on display, in the upper left portion of the screen. To bring a particular scene into the main scene window, the student selects the corresponding terminal node in the map.

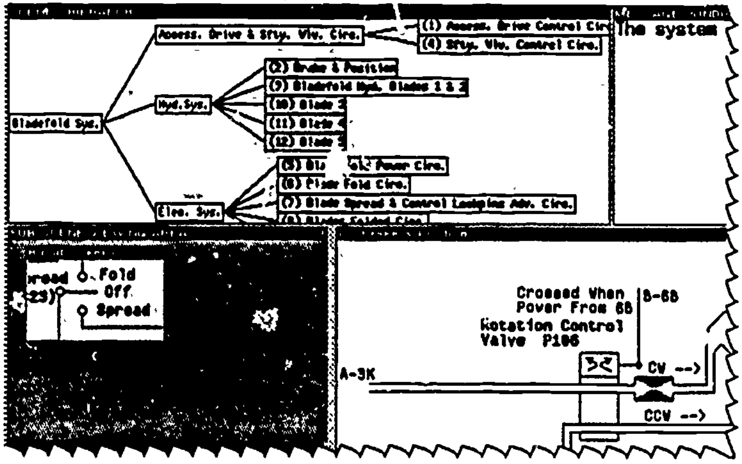


Figure 6. The Scenes Hierarchy and Special Objects Panels

Students or instructors can create an ad hoc scene containing objects of particular interest at any time by selecting the objects from their main scenes and copying them into the gray panel at the left of the main scene window. The objects in this scratch area can be manipulated exactly as are the originals, and they respond graphically to all student actions made either on the ad hoc scene or upon the main scenes. This feature also facilitates the demonstration and examination of interactions of objects that belong to different scenes.

Authoring the Functional Model

This section will briefly outline the simulation construction process within the IMTS. A more thorough description of the IMTS simulation composition system is presented in Towne, Munro, Pizzini & Surmon (1987).

Authors build graphical simulations using a series of four editors. A *graphical editor* is used to draw new types of components, whenever the required component type is not already present in a library. After drawing a component, a *behavior editor* is used to specify how a component operates, and how it can be connected to other components.

The *scene editor* is used to create and to connect instances of the generic component types (an instance of an object is 'created' by simply selecting a generic object type from the library and assigning the object a more specific name). Finally, a *scene connection editor* is used to indicate how the scenes of simulation are interconnected. The scene connection editor also creates links between the elements of the scene map and the actual scenes that are accessed by clicking on those elements.

The graphical object editor and the object behavior editor are currently being merged into one integrated object creation editor. Likewise, the scene editor and scene connection editor are being merged into a single multi-scene simulation creation editor.

In some important respects the process of creating IMTS simulations is similar to that used in STEAMER (Hollan, 1983; Hollan, Hutchins, & Weitzman, 1984), which also makes use of graphical editors for objects. A crucial difference between the two approaches is that in IMTS, system level behaviors are derived from object behaviors, whereas in STEAMER the simulation is produced via conventional computer programming techniques, and the generic objects portray values within the simulated system.

The Object Graphics Editor. The object graphics editor allows construction of objects using graphical primitives such as line segments, rectangles, ovals, and text elements. Figure 7 illustrates the editor in use, expanding a library containing functional representations of objects. Each object shown is fundamentally unique, although some may appear similar. Additionally, most of the objects are multi-state devices, although each is shown in only one graphic state in Figure 7. The toggle switch in the upper left-hand corner, for example, is shown in the Up position. The switch below it is also a two-state switch, but differs in the number of poles available.

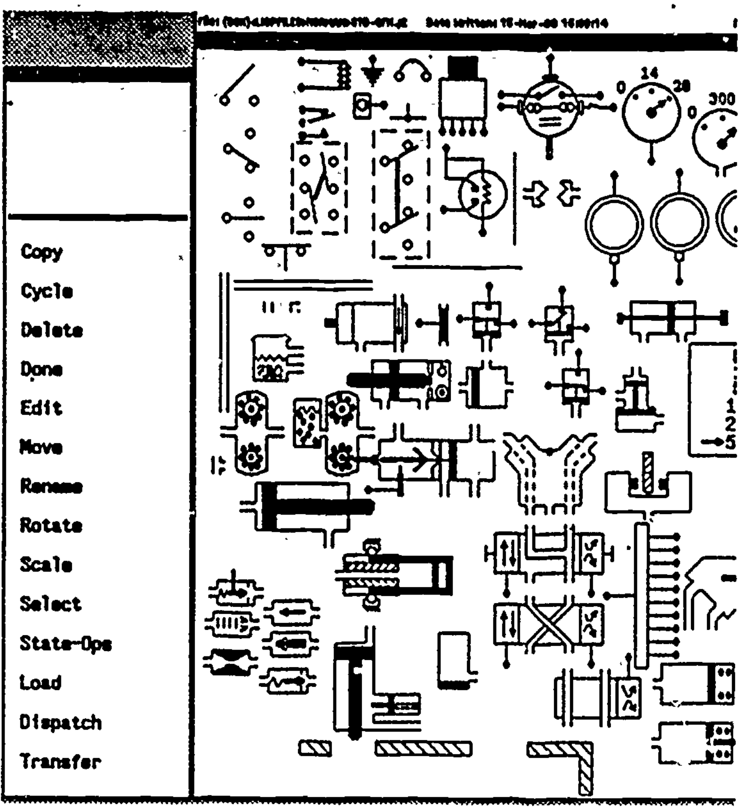
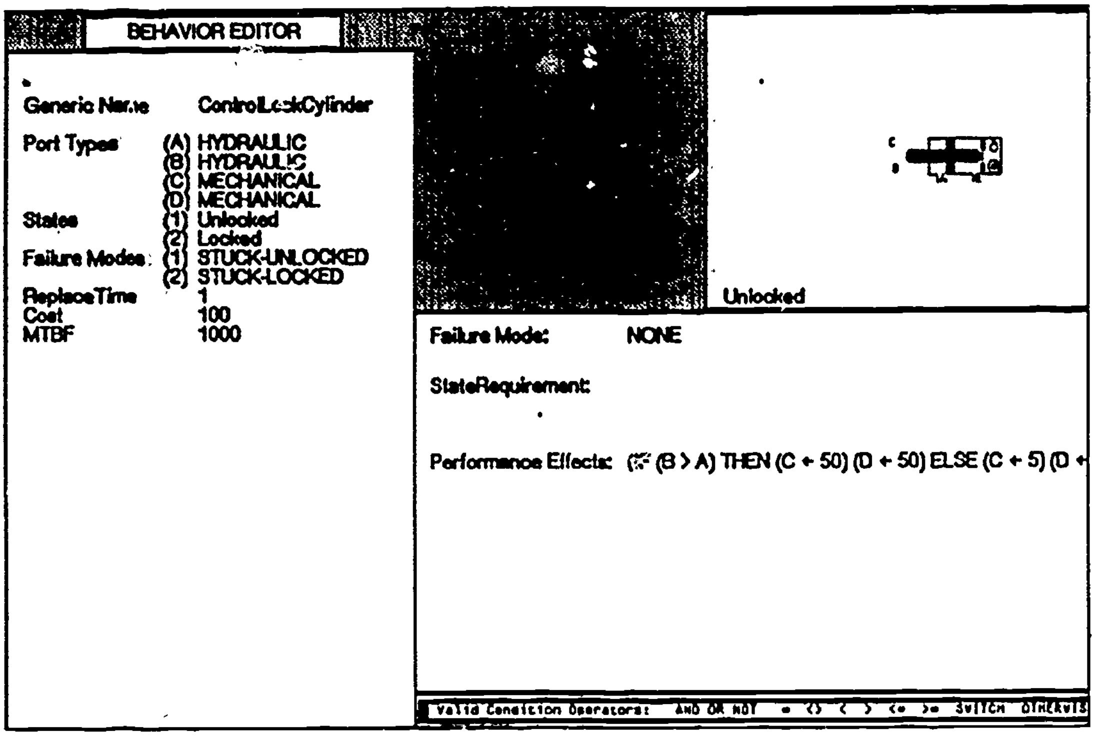


Figure 7. A Library of Functional Object Graphics

The Object Behavior Editor. The object behavior editor is used to describe how a component operates in each state. In this editor, the user enters rules that state the conditions under which the object will take on each state and what output values it will propagate to the outside world, as a function of its inputs. The state-rules for an object may be functions of the previous state of the object, values of its inputs, and relations among its inputs (such as input x > input y).

Figure 8 shows this editor being used to define the operation of a two-state, hydraulically-actuated, locking device.



BEHAVIOR EDITOR

Generic Name ControlLockCylinder

Port Types (A) HYDRAULIC
(B) HYDRAULIC
(C) MECHANICAL
(D) MECHANICAL

States (1) Unlocked
(2) Locked

Failure Modes: (1) STUCK-UNLOCKED
(2) STUCK-LOCKED

ReplaceTime 1
Cost 100
MTBF 1000

Unlocked

Failure Mode: NONE

StateRequirement:

Performance Effects: (IF (B > A) THEN (C + 50) (D + 50) ELSE (C + 5) (D +

Valid Condition Operators: AND OR NOT = <> < > <= >= SWITCH OTHERWIS

Figure 8. The Behavior Editor in Use

The behavior editor automatically generates Lisp code for the user-supplied behavior rules, compiles the code, and adds the resulting function to the object library. The IMTS simulation driver routine executes these compiled routines during training to maintain an accurate graphical representation in response to the student's actions.

The Simulation Editor. The graphic scenes representing sections of the real device are composed using the simulation editor. Authors select generic component types from the libraries and position them on the screen, as shown in Figure 5. When a new component is juxtaposed to another, adjoining ports are automatically connected, i.e., outputs from one component are automatically recorded as flowing to the input of the other. The simulation scenes can be arbitrarily complex.

The Scene Composition Editor. After producing the individual scenes, the simulation developer creates a map of the scenes, previously shown in Figure 6.

The Physical Model

The primary vehicle operated upon by the student to practice diagnostic actions is the physical representation of the target system.

Adjunct Configuration

In the adjunct configuration the physical representation is provided in the form of videodisk images. A touch panel mounted over the videodisk screen allows the student to change a switch setting by touching the desired new position. Likewise, a test point may be read, after selecting a test equipment, by touching a displayed test point and observing the resulting display of the test equipment face. The responses of the target device are computed by GMTS, employing its condition-evaluation technique (a rule-based form), rather than the device model approach.

Stand-alone Configuration

In the stand-alone configuration, the IMTS functional model is used to determine system behaviors, and both the functional and the physical representations are updated to reflect these computed responses. In this configuration the physical model may be represented with either videodisk images or computer graphic images. The graphic form of physical objects is created using the object graphic editor as shown in Figure 9.

The graphic views of front panels and other hardware sections are created by building scenes in which the physical objects are simply tied, or slaved, to their counterparts in the functional scenes. The slaved physical objects therefore react correctly by simply referring to the underlying functional model. A physical scene can therefore be constructed in which the objects appear to be connected to each other, but in fact these apparent connections are not involved in determining system behavior.

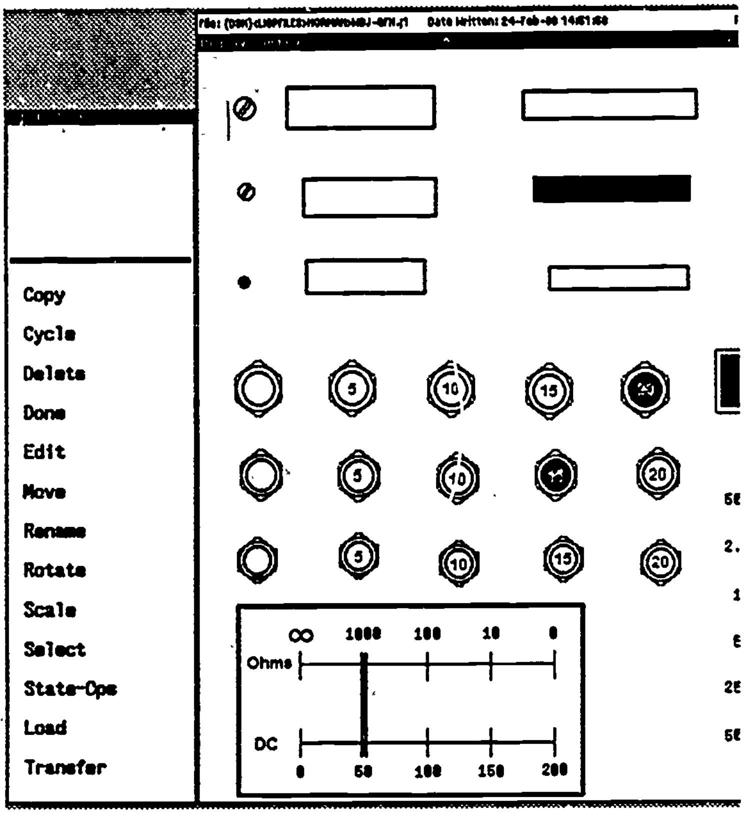


Figure 9. Objects in a Physical Representation

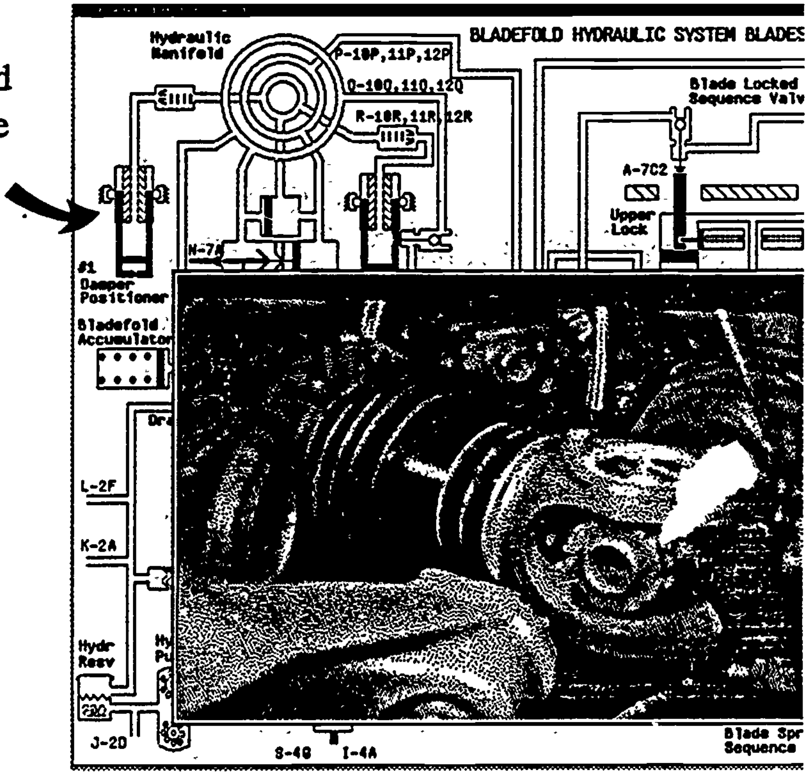
Links Between the Physical and Functional Representations

An important concern in designing the IMTS was providing linkages between the physical and functional representations, so that students would experience frequent and memorable exposures relating the two. The intention is to promote the student's ability to find, recognize, and manipulate physical elements in the real system, while maintaining a conception of the functional relationships that cannot be seen directly in the real system.

The techniques to do this involve 1) providing both physical and functional representations of the system so that students can examine equivalent views either sequentially or simultaneously, and 2) providing both physical and functional views of single objects.

Physical and Functional System Representations. Because physical objects are linked to their functional counterparts, for the purpose of supporting the physical simulation, the IMTS will be able to automatically alternate between the two, at the request of the student. Additionally, the videodisk representation can follow along automatically as the student operates in the functional form. Of course functional organization is not likely to correspond with physical structure, therefore the physical scene corresponding to one functional object might be different than that of a neighbor object, and vice versa.

Physical and Functional Views of Individual Objects. To strengthen the student's understanding of physical and functional equivalence, the IMTS can display 'pictures' of individual objects within their functional context. In Figure 10 the student was viewing a functional scene and was curious about the appearance of the #1 Damper Positioner (near the upper left). Upon requesting a pictorial view of the part, the student sees the presentation shown.



This object is the selected one that is depicted in the bitmap picture

Figure 10. Bladefold Scene with a Digitized Picture

SECTION III. Problem Selection and Student Modeling

In the current IMTS, the student model is used primarily for the automatic selection of appropriate problems for individual students. In future versions the current state of the student model may play a role in determining other aspects of instruction and aiding, such as frequency of intervening.

Normative Student Modeling

IMTS uses an expert, or normative, model to represent device understanding for diagnostic functions. This hierarchical model of domain knowledge is created by the device expert, and used as the default model of each individual student. As the student performs troubleshooting problems, the node weights of the model are adjusted based on the quality of the problem solving work. This approach is loosely based on the student modeling mechanism employed in BIP (Wescourt, Beard, & Gould, 1977; Wescourt & Hemphill, 1978; Wescourt, Beard, Gould, & Barr, 1977; Beard, Barr, Gould, & Wescourt, 1978) in which a model of student knowledge and skill was based on a component analysis of the content of a curriculum. The curriculum content was represented as a set of elementary concepts and skills. Each instructional exercise or problem was considered diagnostic for a particular subset of these concepts and skills. Student performance on the problems was used to modify a student-specific model of knowledge in the domain.

The model has the form of a tree in which each node represents the knowledge about some aspect of the equipment. The highest nodes in the tree represent the most abstract knowledge about the global functions of the equipment. The nodes immediately below that represent top-level knowledge about the major subsystems. Lower nodes represent more specific skills and knowledge about modules and components. The terminal nodes in the knowledge tree represent knowledge about specific component modes, including different possible types of component failures.

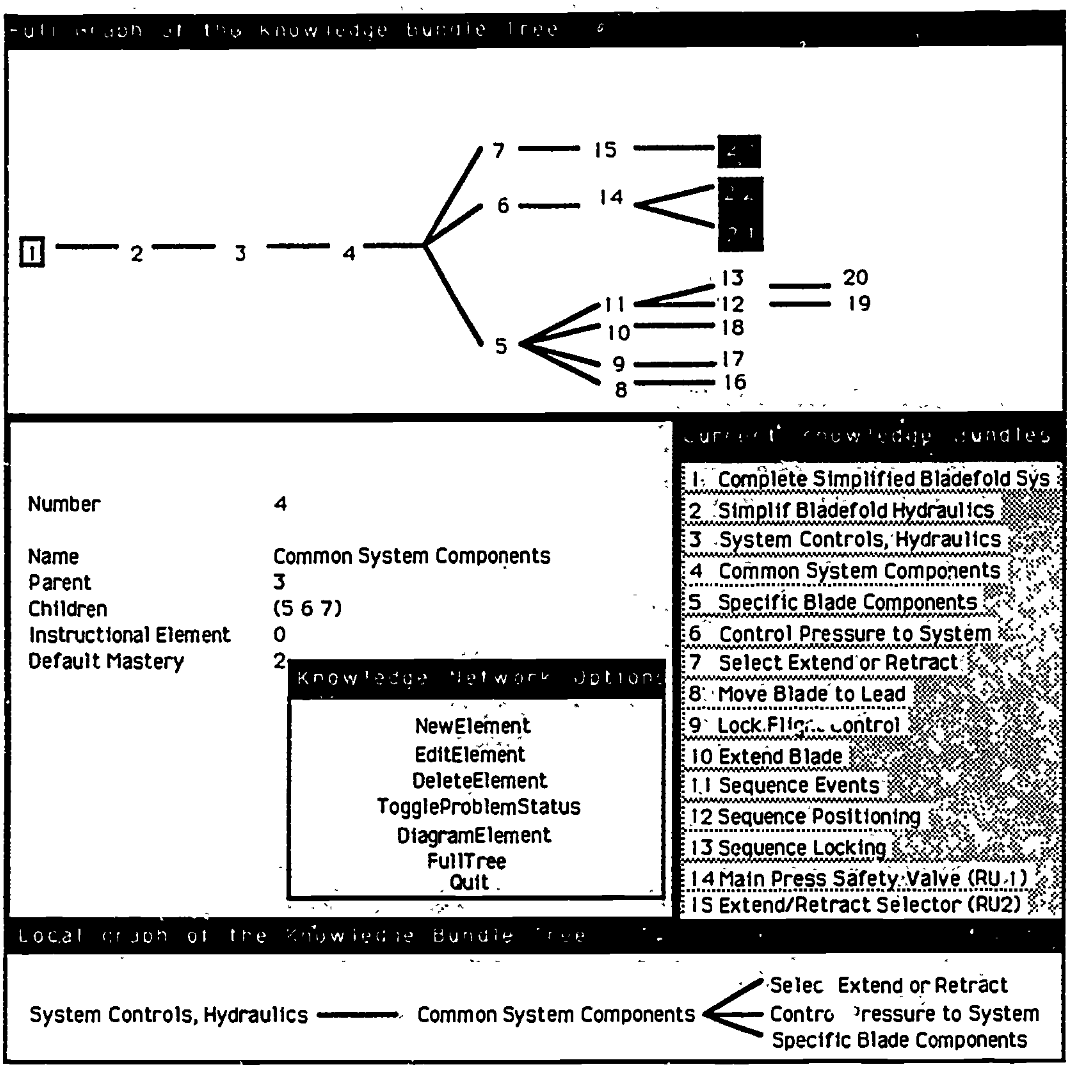
The model is linked closely to the structure of the device being simulated. This has the advantage of making the creation of such models fairly straightforward, and offers the potential for making the normative modeling process automatic. A disadvantage is that generic kinds of knowledge, such as Ohm's law, are not easily represented in such an equipment-specific knowledge model.

Creating the Normative Model

The normative knowledge tree is built using the IMTS knowledge network editor (see Figure 11). The current implementation of the Knowledge Network Editor makes very few assumptions about the structure of knowledge in device domains. (Enhancing the editor to capitalize on domain-specific features is a likely topic for future research and development efforts.) Providing structure is the task of the knowledge network author. One assumption of the Editor is that device knowledge can be represented in a simple tree structure. Less specific, more global information is represented by nodes near the top or root of the tree. Detailed information, such as the particular failure modes of particular elements in the device, are represented by nodes at the bottom of the tree.

Maintaining an Individual Student Model. The understanding of individual students is represented by a set of weights for the nodes of the tree. These weights represent how well the student understands the corresponding portions or functional subsystems of the device.

When a student finishes a problem, a measure of his or her performance for that problem is computed to modify the student model. The node that corresponds to the actual malfunction is directly assigned the value of the performance measure. Each of the ancestor nodes for that node is also modified to reflect the performance as well. The immediate parent is modified in the direction of the performance by an amount weighted by the number of child nodes that it has. Its parent node is similarly modified by an amount relative to the extent of the change in that node, and so on up the ancestor tree. A change in a single specific malfunction node can have a (usually small) effect even on the top level node, which represents the student's understanding of the whole system.



Number	4
Name	Common System Components
Parent	3
Children	(5 6 7)
Instructional Element	0
Default Mastery	2

1 Complete Simplified Bladefold Sys
2 Simplif Bladefold Hydraulics
3 System Controls, Hydraulics
4 Common System Components
5 Specific Blade Components
6 Control Pressure to System
7 Select Extend or Retract
8 Move Blade to Lead
9 Lock Flight Control
10 Extend Blade
11 Sequence Events
12 Sequence Positioning
13 Sequence Locking
14 Main Press Safety Valve (RU-1)
15 Extend/Retract Selector (RU2)

Figure 11. The Knowledge Network Editor

Problem Selection

Problem selection is made on an individual basis, using the proficiency records for the student, as encoded in the student model, and the global measures of student ability and cognitive style. Proficiency is measured by normalized time to solve the previous problem and on average test power for the exercise. The model is updated at the conclusion of each problem, to support the selection of appropriate subsequent problems for students.

The problem selection process relies on two measures taken from the knowledge structure: conceptual distance and conceptual difficulty. Conceptual distance is a simple measure of how related two problems are in terms of the domain representation. The conceptual distance between two problem nodes is the number of node links that must be traversed in the knowledge element hierarchy to find one node from another, weighted by the student's current mastery levels for the intervening nodes. Conceptual difficulty is the value of the "Problem Difficulty" field created by the instructor.

To select a new problem, the remaining troubleshooting exercises are evaluated for their conceptual distances from the last problem and their difficulty. An ideal conceptual step size and an ideal difficulty for the next problem are then computed for the student. The desired conceptual step size is a function of the student's estimated learning speed (which is based on prior performance and an instructor estimate), a student-controlled value that expresses how large a conceptual jump the student likes to make, and the student's performance on the last problem.

- If the student has a high learning speed, conceptual distance can be larger.

- If the student prefers larger conceptual steps, the conceptual distance can be larger.

- If the student did well on the last problem, the conceptual distance can be larger. (If the student did poorly on the last problem, a related one is called for.)

- The ideal difficulty level for the student is a function of the student's learning speed.

It is rare that the ideal conceptual distance metric and the ideal difficulty metric pick the same problems. A weighting scheme combines these factors.

Dimensions of Troubleshooting Problem Difficulty

One of the goals of simulation training is to provide practice at an appropriate level of difficulty. If students are presented with problems that are too easy, they will simply reapply old skills and learn nothing new. On the other hand, if problems are too difficult, students may become discouraged and fail to learn. To a large extent, the difficulty of a troubleshooting problem depends on the idiosyncratic knowledge of a student. One student of a helicopter bladefolding system may be very familiar with the hydraulic components responsible for positioning the No. 1 blade, while another is more cognizant of the subsystems related to the rotor brake. These two students would not order a set of problems for difficulty in the same way.

The major components of problem difficulty in IMTS include
- the size of the initial suspicion set
- the level of representation required to discriminate the fault
- the knowledge of detailed component behavior required
- the remoteness of symptoms from the suspected faults

IMTS offers several means of presenting a fault isolation exercise so as to reduce or increase its difficulty. These means include managing the instructional and aiding features described in Section IV. In addition, it is possible to manipulate the complexity of the device representation.

Controlling the Complexity of the Representation. Using the same approach employed for constructing physical representations, simulation developers may create alternate forms of representations in which the objects are simply slaved to the fully detailed functional model. As a result, highly simplified diagrams can be produced that still exhibit accurate behaviors. For students who are unfamiliar with the helicopter bladefold system, for example, a simplified simulation can be constructed showing only one blade and omitting auxiliary parts that are not central to the operation of the device. The simplified forms may be either functional or physical.

While the problem selection process does not yet consider moving to problems in alternative representations, it will be enhanced to do so when the complete authoring capabilities have been completed.

SECTION IV. IMBEDDED GENERIC EXPERTISE

Two kinds of generic expertise are imbedded in IMTS, 1) diagnostic expertise and 2) instructional expertise. The diagnostic expertise, in the form of the Profile fault-isolation process, is called frequently to assess the student's work and to provide technical advice and explanations. The instructional strategy is embodied in executive routines that continually assess student progress and call on the specialized functions to respond to the student's needs. We first describe the generic diagnostic expert model.

Diagnostic Expertise from Device Data

An important goal of the IMTS was to generate the diagnostic intelligence for a particular device from the same functional model that supports the graphical representation. This leads to high efficiency of course development, and it ensures that the diagnostic expert and the student operate upon the same system model. As a result, all diagnostic advice provided can be explicitly rationalized to the student.

To identify, justify, and interpret a good next test, Profile requires data specifying the significance of each possible symptom for each possible test, in each mode of interest (a mode is a combination of switch settings). This is turn requires that the device be simulated in each failure condition to determine what symptoms are produced in each specified mode. Because this involves considerable compute time, the computation of symptoms is done during the development phase, and the results are stored in a data file for quick access during training.

The utility program that produces the symptom data inserts a failure into the device model, it runs the IMTS simulation program in all the modes of interest, as specified by the course developer, and it records the results at each indicator and test point. By comparing each result to that obtained with no failure inserted, the program can identify all abnormal symptoms resulting from the failure. This process is repeated for each possible failure.

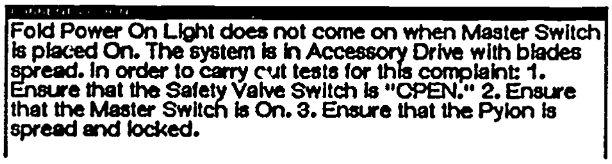
While the resulting data file is large, only limited portions are accessed for any one practice problem. The complaint-specific data are read in at the beginning of each problem, and remain accessible throughout the problem. When Profile is called upon to suggest a test early in a problem, when there are many alternative tests and suspected elements, it requires between five to ten seconds to complete its analysis of the alternatives. If the student has performed even one or two useful tests, the Profile compute time drops off rapidly to just a few seconds.

Required Human Expertise

As mentioned above, human intelligence is required to specify the modes to be considered in computing fault effects. To not constrain this variable would lead to impossible data file sizes and compute times. The training implication of this restriction is that the IMTS can only monitor and advise the student as long as he or she works in one of the recognized modes. Preliminary experience indicates that this limitation is not particularly distressing, as each problem has a natural and relatively limited set of modes appropriate for fault diagnosis.

Human knowledge is required to provide a second element, a failure report for each practice problem that states the general nature of the problem as it would be reported by the equipment operator. These statements typically list one or more high-level system functions that do not operate normally, and some conditions, such as the mode, under which the abnormality was observed.

The failure report is displayed at the beginning of each problem, and may range from highly informative to quite vague. The course developer has the option of issuing the same failure under different failure reports, thereby varying the difficulty of the problem.



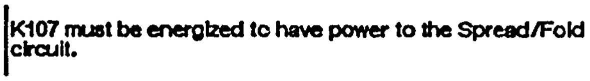
Fold Power On Light does not come on when Master Switch is placed On. The system is in Accessory Drive with blades spread. In order to carry out tests for this complaint: 1. Ensure that the Safety Valve Switch is "OPEN." 2. Ensure that the Master Switch is On. 3. Ensure that the Pylon is spread and locked.

Figure 12. An Initial Failure Report

Options for Adding Human Knowledge

There are two options for enhancing each practice problem by adding human knowledge, 1) a hint to help the student get started on the problem, and 2) a technical summary that explains the physical (electronic, hydraulic, etc.) mechanisms of each practice failure. Each of these may be extensive, brief, or omitted entirely.

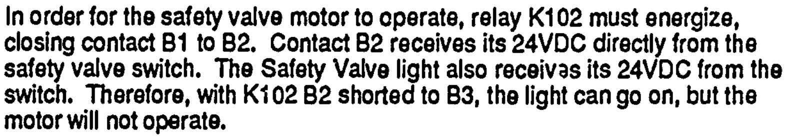
Problem Hint. If provided, the hint is automatically presented if a student is foundering early in the problem and has not requested assistance from the IMTS. The hint is associated with the failure report, thus one hint may support presentation of many problems.



K107 must be energized to have power to the Spread/Fold circuit.

Figure 13. A Pre-Authored Hint

Failure summary. If available, the failure summary is presented upon completion of a problem to more fully explain why and how the symptoms were produced. Although the simulation routine generated the symptoms, by applying and propagating each object's behavior rules, human-supplied explanations provide the most rich and meaningful account of the technical nature and consequences of the simulated failure. A typical failure summary is shown in Figure 14.



In order for the safety valve motor to operate, relay K102 must energize, closing contact B1 to B2. Contact B2 receives its 24VDC directly from the safety valve switch. The Safety Valve light also receives its 24VDC from the switch. Therefore, with K102 B2 shorted to B3, the light can go on, but the motor will not operate.

Figure 14. A Failure Summary

Generic Instructional Expertise

Students have partial control of the aiding features described below. These instructional options will be presented automatically to students who need them, but, in addition, they can be requested by students.

Several types of help are available, in addition to the simple hints mentioned above. The help functions range in purpose from discussing a single symptom to direct instructions about what course of action to follow next. These levels of aiding make it possible to help a student while still preserving some potential for student discovery. Those students who cannot make it through a difficult troubleshooting problem on their own can still solve, and learn from, a problem through the use of the more directed aiding features. (Currently, the student model does not take the degree of assistance provided into account when updating the student's proficiency; this will be incorporated in the future.)

The IMTS uses four different styles of aiding and instructing the student, all based on the generic troubleshooting expertise of Profile in exploring and applying the fault effect data.

Within-problem Monitoring and Assistance

Student performance is monitored within problems to determine what help or instruction is required to minimize unproductive time and to help ensure success on the problem. Two major elements of the student's performance are tracked within a problem: time on the problem and average test power per unit of time on the problem. The measure of time spent on a problem is a relative one; time on the troubleshooting problem is compared with a parameter that compensates for the difficulty of that problem. The average test power measure is more interesting. Profile determines the relative power of each test performed by a student by comparing it to the one Profile would have performed at that point. The average of these values provides a good assessment of a student's diagnostic performance on a problem.

Parameters ranging from zero to one may be entered by an instructor in order to specify the degree to which students may deviate from ideal performance before receiving assistance. Four different test power parameters can be set. The first of these determines at what test power threshold the student will be offered instructional help for the first time. A second parameter determines at what threshold subsequent offers of help will be made. The third parameter sets the test power value at which assistance will be given whether the student wants it or not, and the fourth prescribes the test power level for subsequent enforced assistances. A value of 0.5 permits significant departures from maximum test power before overt aiding or instructional features are brought into play.

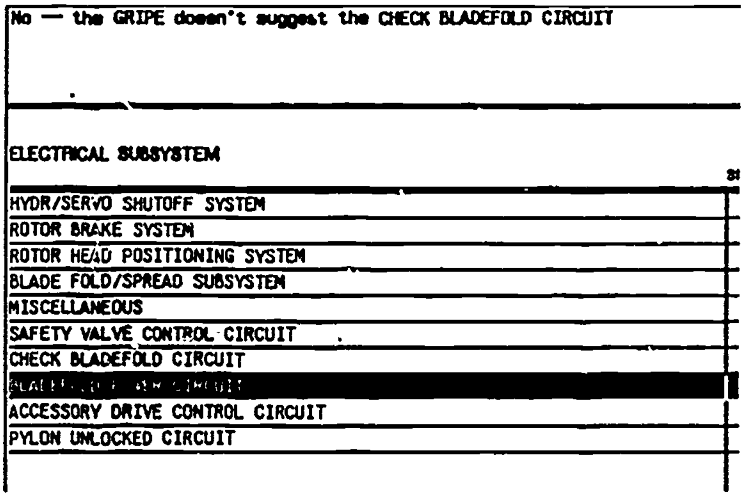
A prime objective of IMTS is to operate in an unobtrusive manner, allowing the student to practice fault isolation in a more realistic context than one in which advice continually intrudes on the troubleshooting process. The IMTS remains largely invisible to the student until the student requests aid or until the monitoring process reveals that the student is having difficulty. The indications of difficulty include taking too much time relative to the norms for a problem and making too many tests of low power. Possible causes of low diagnostic productivity include misinterpretation or misapplication of an earlier test result or inability to identify a test that has potential in isolating the source of the problem.

Eliciting (and Refining) a Student's Suspicions. There are occasions during training when assessment of student actions requires knowing the student's beliefs or intentions. The major means of ascertaining the student's beliefs about the malfunction state is to use the IMTS suspicion eliciting system. The student is posed a number of questions about what portions of the device he or she suspects. The student's responses are corrected until a set of suspicions has been agreed upon that is correct *based on the information thus far available*. At this point the student and IMTS have collaborated to identify a set of suspicions that make sense, given the test results that have been seen.

This process relies upon a representation of the subject device as a hierarchy of functional subsystems. In the case of the helicopter bladefold system, for example, the bladefolding mechanism was analyzed as consisting of, first, a hydraulic and an electrical subsystem. Each of these was treated as having a number of major descendent subsystems, and so on. At the lowest level in this analysis are particular failures; just above this level are replaceable components. This analysis of the higher level structure of the device is similar in many respects to the normative student model created with the knowledge network editor described in Section III. In a future version of IMTS, the same data structure will fill both roles — the normative knowledge model and the hierarchical structure of the device used to discuss suspicions.

The first time that a student's suspicions are elicited, the IMTS asks whether the student suspects any of the highest level subsystems. In the case of the Bladefold system, the student would be asked whether anything should be suspected in the electrical system and whether anything should be suspected in the hydraulic system. If the symptoms obtained by the student justify reducing the suspected set to only one of the subsystems, then the process of eliciting suspicions continues by inquiring about each of the subsystems of *that* system.

Figures 15 and 16 present the suspicion eliciting system of IMTS in action. The panel in Figure 15 lists the subsystems that should be considered at any particular level of system organization. The routine steps through this list, highlighting the subsystem currently in question.



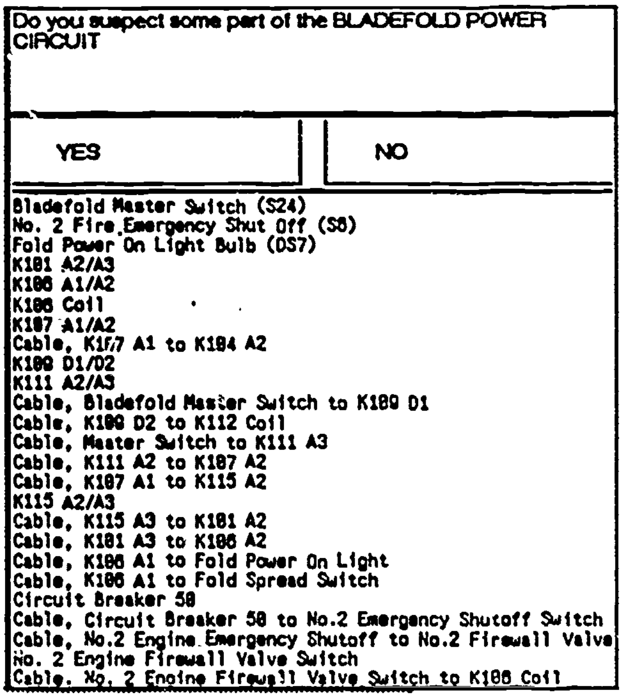
No — the GRIPE doesn't suggest the CHECK BLADEFOLD CIRCUIT

ELECTRICAL SUBSYSTEM

HYDR/SERVO SHUTOFF SYSTEM
ROTOR BRAKE SYSTEM
ROTOR HEAD POSITIONING SYSTEM
BLADE FOLD/SPREAD SUBSYSTEM
MISCELLANEOUS
SAFETY VALVE CONTROL CIRCUIT
CHECK BLADEFOLD CIRCUIT
[illegible]
ACCESSORY DRIVE CONTROL CIRCUIT
PYLON UNLOCKED CIRCUIT

Figure 15. A Portion of the IMTS Suspicion Eliciting Mechanism

Figure 16 shows the student being asked about his suspicion of the Bladefold Power Circuit, and being shown the components in this subsystem, for reference.

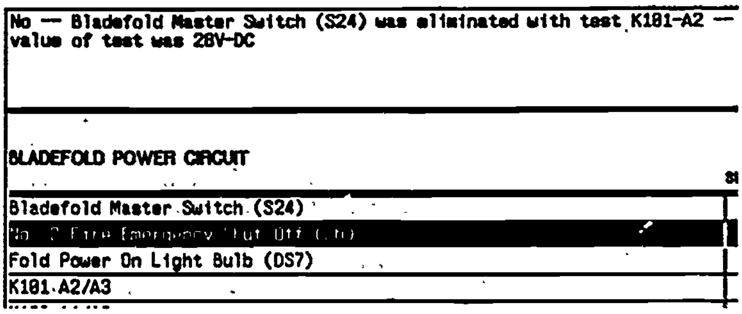


Do you suspect some part of the BLADEFOLD POWER CIRCUIT

Bladefold Master Switch (S24)
No. 2 Fire Emergency Shut Off (S8)
Fold Power On Light Bulb (DS7)
K101 A2/A3
K106 A1/A2
K106 Coil
K107 A1/A2
Cable, K107 A1 to K104 A2
K109 D1/D2
K111 A2/A3
Cable, Bladefold Master Switch to K109 D1
Cable, K109 D2 to K112 Coil
Cable, Master Switch to K111 A3
Cable, K111 A2 to K107 A2
Cable, K107 A1 to K115 A2
K115 A2/A3
Cable, K115 A3 to K101 A2
Cable, K101 A3 to K106 A2
Cable, K106 A1 to Fold Power On Light
Cable, K106 A1 to Fold Spread Switch
Circuit Breaker 50
Cable, Circuit Breaker 50 to No.2 Emergency Shutoff Switch
Cable, No.2 Engine Emergency Shutoff to No.2 Firewall Valve
No. 2 Engine Firewall Valve Switch
Cable, No. 2 Engine Firewall Valve Switch to K106 Coil

Figure 16. The Details Panel in Suspicion Eliciting

The IMTS keeps track of the significance of tests as they are performed by the student, remembering which suspected elements could have been eliminated by each. It can therefore tell a student why a particular element should no longer be suspected. If a student suspects an element when a previously-conducted test should have eliminated it from suspicion, then IMTS reminds the student of that test and the value it revealed (see Figure 17). Here the student has just indicated that he suspects the Bladefold Master Switch. IMTS points out that a previously seen test result should have eliminated this circuit from suspicion.



No — Bladefold Master Switch (S24) was eliminated with test K101-A2 — value of test was 28V-DC

BLADEFOLD POWER CIRCUIT

Bladefold Master Switch (S24)

No. 2 Fire Emergency Shut Off ([illegible])

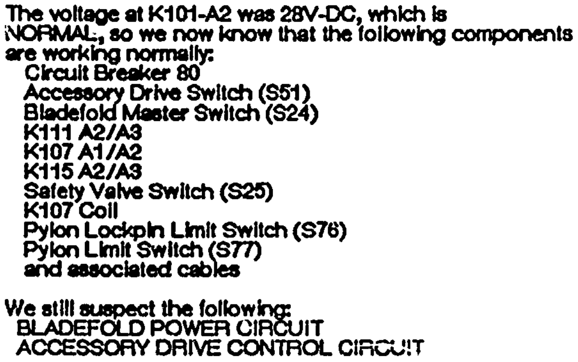
Fold Power On Light Bulb (DS7)

K101 A2/A3

Figure 17. Refining Student Suspicions

Explaining the Meaning of the Most Recent Test. The elicitation and remediation of beliefs about possible failures considers all previous tests performed by the student on that problem. Sometimes the student may need assistance in interpreting a particular symptom. He or she may request that IMTS interpret the symptom when it is displayed.

In response to the request, the IMTS first recapitulates the observation and states whether the observed value is normal or abnormal. Based on that judgement, it lists the subsystems or objects that should still be suspected and those that should no longer be suspected.



```
The voltage at K101-A2 was 28V-DC, which is
NORMAL, so we now know that the following components
are working normally:
  Circuit Breaker 80
  Accessory Drive Switch (S51)
  Bladefold Master Switch (S24)
  K111 A2/A3
  K107 A1/A2
  K115 A2/A3
  Safety Valve Switch (S25)
  K107 Coil
  Pylon Lockpin Limit Switch (S76)
  Pylon Limit Switch (S77)
  and associated cables

We still suspect the following:
 BLADEFOLD POWER CIRCUIT
 ACCESSORY DRIVE CONTROL CIRCUIT
```

Figure 18. Explaining Diagnostic Test Results

Suggesting Troubleshooting Actions. Sometimes a student requires very direct assistance in proceeding on a problem. In general, this could be termed a 'planning' skill, although in diagnosis the scope of future considerations is normally limited to just identifying the next best action.

It could be argued that in such a case the student's instructor should be notified so that the student can receive individual attention and instruction. In those cases in which instructor time is largely committed, this level of individualized instruction may be difficult to guarantee. Even if the instructor were always available, however, it is likely that a mandatory interruption in the student's practice problem for instructor remediation could quickly come to be viewed as a signal of failure.

An alternative approach is to provide mechanisms that assure that the student can never reach a complete impasse. Recent research (Fox, 1987) suggests that human tutors try to prevent failure. The IMTS approach to doing this on the troubleshooting task is to provide the student with suggested troubleshooting actions that are guaranteed to be useful and rational (and are actually near-optimal). Figure 19 shows the form of such advice.

```
Measure the voltage at K101-A2
28V-DC would be normal.

If normal, the failure is one of:
 K101 A2/A3
 K106 A1/A2
 K106 Coil
 Fold Power On Light Bulb (DS7)
 Circuit Breaker 50
 No. 2 Engine Firewall Valve Switch
 No. 2 Fire Emergency Shut Off (S6)
 Linear Actuator
 and associated cables

If abnormal, the failure is one of:
 Circuit Breaker 80
 Accessory Drive Switch (S51)
 Safety Valve Switch (S25)
 Bladefold Master Switch (S24)
 Pylon Limit Switch (S77)
 K111 A2/A3
 K107 A1/A2
 Pylon Lockpin Limit Switch (S76)
 K115 A2/A3
 K107 Coil
 and associated cables
```

Figure 19. Generated Troubleshooting Suggestions

By exploring the fault-effect data, following each test performed by the student, Profile identifies the next test that will best discriminate among the current suspicions. This process is done whether or not the student requests assistance. If the student proceeds without getting help, then Profile simply compares the student's test to its own selection, to yield a test power measure. If the student requests assistance in proceeding, then Profile presents and explains its selection.

Between-problem Instruction

Because a fault isolation exercise involves an unknown to be discovered by the student, the IMTS attempts to defer some remediations and explanations — those that would destroy the value of the problem — until after the problem has been completed. At that time, a number of choices are offered for further exploring the now-known failure, as shown in Figure 20.

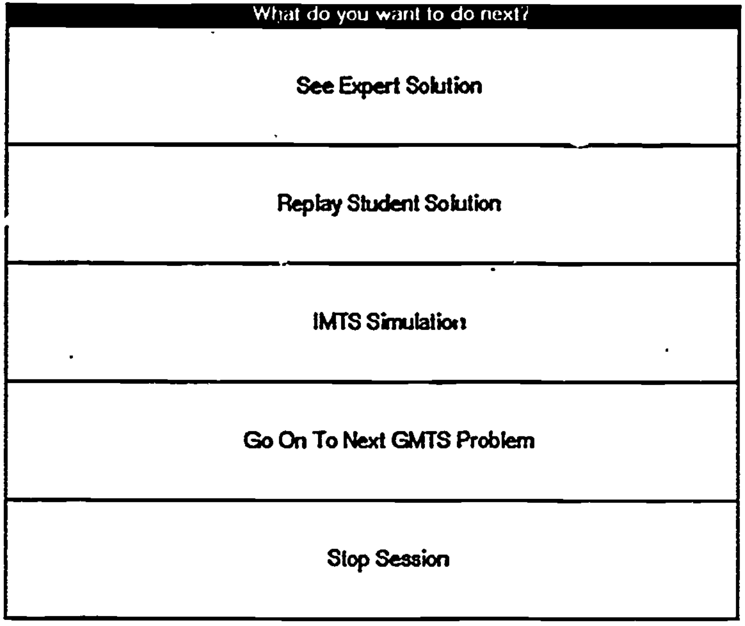
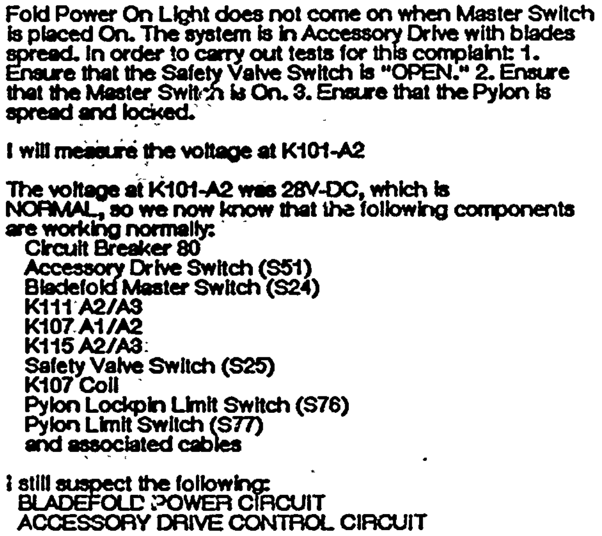


Figure 20. The Student's Choices Between Problems

These choices include two instructional options that, like those already discussed, make use of the Profile analysis of the troubleshooting context: presentation of an expert solution and a critiqued replay of the student's solution.

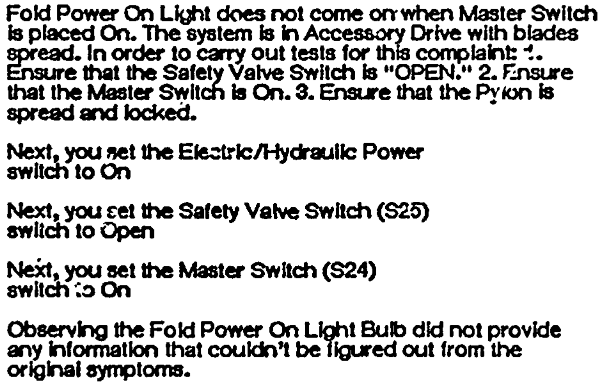
Expert Troubleshooting of the Same Problem. The IMTS describes each action that Profile dictates for troubleshooting the reported fault and it interprets the results that would be seen for each. The interpretation lists the elements eliminated from suspicion by the test and those for which suspicion should increase as a result of the test. Figure 21 shows the start of such an expert solution of a problem.



```
Fold Power On Light does not come on when Master Switch
is placed On. The system is in Accessory Drive with blades
spread. In order to carry out tests for this complaint: 1.
Ensure that the Safety Valve Switch is "OPEN." 2. Ensure
that the Master Switch is On. 3. Ensure that the Pylon is
spread and locked.

I will measure the voltage at K101-A2

The voltage at K101-A2 was 28V-DC, which is
NORMAL, so we now know that the following components
are working normally:
   Circuit Breaker 80
   Accessory Drive Switch (S51)
   Bladefold Master Switch (S24)
   K111 A2/A3
   K107 A1/A2
   K115 A2/A3
   Safety Valve Switch (S25)
   K107 Coil
   Pylon Lockpin Limit Switch (S76)
   Pylon Limit Switch (S77)
   and associated cables

I still suspect the following:
  BLADEFOLD POWER CIRCUIT
  ACCESSORY DRIVE CONTROL CIRCUIT
```

Figure 21. Presentation of a Generated Expert Troubleshooting Sequence

Debriefing the Student with a Problem Replay. Another option offered to students at the end of a troubleshooting problem is to review their own sequence of actions, with generated commentary. This presentation describes what the student should have learned from each test that was performed, and identifies tests that had no value in the context of the troubleshooting sequence followed. See Figure 22.



```
Fold Power On Light does not come on when Master Switch
is placed On. The system is in Accessory Drive with blades
spread. In order to carry out tests for this complaint: 1.
Ensure that the Safety Valve Switch is "OPEN." 2. Ensure
that the Master Switch is On. 3. Ensure that the Pylon is
spread and locked.

Next, you set the Electric/Hydraulic Power
switch to On

Next, you set the Safety Valve Switch (S25)
switch to Open

Next, you set the Master Switch (S24)
switch to On

Observing the Fold Power On Light Bulb did not provide
any information that couldn't be figured out from the
original symptoms.
```

Figure 22. Problem Replay with Generated Commentary

SECTION V. CONCLUSIONS

Findings and Recommendations

In the course of this research and development project, we have explored a number of alternative approaches to simulation, student monitoring, problem selection, and student aiding. In addition to the techniques presented here, other less successful approaches were developed, evaluated, and discarded. To share both the successes and failures, we present the following conclusions.

Device models for simulation-based diagnostic training should support quantitative modeling. To some extent, the object behavior rules in the IMTS appear qualitative. For example, a typical rule states that something happens when a value at one port is larger than the value at another port. This type of expression is effective because that is the way the object actually behaves. In other cases objects respond to the values at ports. For example, certain pressure valves change state when the pressure exceeds some characteristic value. Furthermore, to correctly determine how resistive parts will behave in an electrical circuit, it is necessary to keep track of very precise values within the circuit. As a result, the IMTS simulation program is essentially quantitative.

It is tempting to develop a simulation composition system that does not require that accurate quantitative values be passed from one object to another. At first blush, it appears that both the authoring of object behaviors and the simulation driver software can be made simpler by using a qualitative or 'fuzzy' quantitative approach. Unfortunately, when objects pass imprecise values, two serious problems arise in the simulation. First, it becomes awkward or impossible to provide simulated test equipment. Special-purpose simulated test equipment must be constructed that translates inaccurate or non-quantitative values and converts them into the nominal values expected in the device. This would be necessary if the student is to be able to practice interpreting quantitative readings back to qualitative judgements such as "normal" or "high".

Second, in complex devices, it becomes difficult or impossible to avoid incorrect device responses, owing to inaccuracies in internal effects. In summary, simulations of complex systems tend not to work when the underlying model is approximate.

This is in no way intended to minimize the fascinating work being done with qualitative models. We simply find that complex models may not respond correctly when they operate upon approximate values. Further this finding does not suggest that human diagnosticians operate quantitatively. In fact, the quantitative manipulations that the IMTS simulation program performs, to maintain the graphical simulation, are not apparent to the student.

Visual representations must not be tightly linked to the device model. Instructors and simulation developers need to be able to make simplified presentations that do not involve every element in the real device. Yet, if the graphical objects can obtain their behaviors only from their own underlying functional behavior rules, then every component would have to be included. This awkward restriction led to the development of the feature that allows physical simulations and simplified simulations to obtain their behaviors from objects in the functional model.

Advice does not have to be optimal, but it must be rational. One of the most important aspects of successful training using computer-generated materials is maintaining the trust and confidence of the learner. For complex systems, slight non-optimalities in advice are very unlikely to be noticed in the course of a single student's instruction. In fact, moderate non-optimalities are scarcely a matter of concern, since even excellent human tutors are significantly non-optimal in the troubleshooting strategies they demonstrate. On the other hand, irrational advice is devastating to student confidence. It is much better for an artificial instructor to remain silent than to risk offering truly questionable advice.

Assessment of student performance must be accurate and based upon very recent performance. It is frustrating for a student who has been struggling, but who has just seen the light and begun to make progress in a problem, to suddenly be interrupted with unwanted admonitions and instruction. It is probably better to err on the side of too few interruptions than to interrupt excessively. Further, it has been found crucial to only consider very recent performance, in deciding when to intervene, as lengthy moving averages can trigger interruptions even when the student has just corrected course, and is on the way to a solution.

Accurate tracking of the student conflicts with the goal of a transparent and efficient practice environment. Many of the actions that students take could have several different rational motivations. The number of irrational motivations for a test or replacement is unbounded. In order to really know what the student intends, it is probably necessary to ask. Unfortunately, constantly harassing the student with demands that every action be justified in detail radically changes the nature of the troubleshooting task (and makes it very unpleasant). A balance must be sought to ensure effective training, and the artificial instructor must always be aware that its model of student intent is probably imperfect.

The need to ask the student about beliefs or intentions was part of the motivation for the development of the suspicion eliciting system used in IMTS. Further work is underway to reduce the intrusiveness of this process, so that it can be used more often. This will make it possible to track student intent more closely without damaging the natural troubleshooting process.

Parameters currently used to manage instruction are arbitrary. In the present system, interventions are based upon accurate computations of student performance, compared to quite arbitrary threshold values (such as inactivity exceeding three minutes, or two irrational tests in a row). Ideally, a science of instruction would serve as the basis for determining the appropriate parameters for making decisions about instructional intervention.

Precomputing of fault effects is necessary. For a very complex device, a single fault can have many effects. To discriminate among possible faults, all the effects of each fault must be computed. Although we have made very significant improvements in simulation speed by aggressively pursuing that goal, computed simulations cannot be made instantaneous. Where a large number of faults must be considered, each with a significant number of effects, precomputing these effects is essential to providing advice to students in a timely fashion.

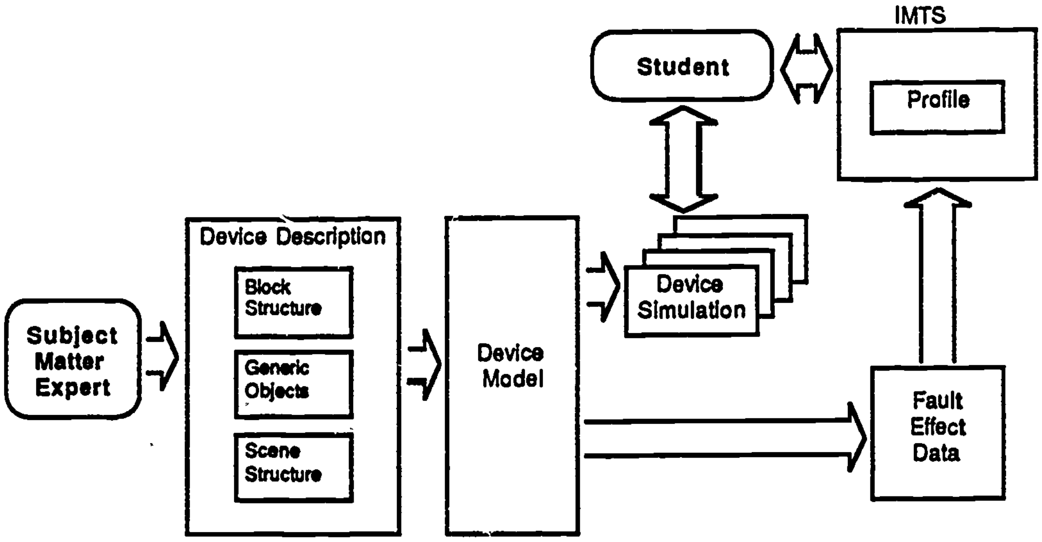
The Next Steps for Enhancing Training in IMTS

IMTS represents a powerful environment for producing and delivering interactive simulation training. It also constitutes a promising foundation on which to build an even more capable technical training system. Here we briefly touch on the features to be implemented in the near future.

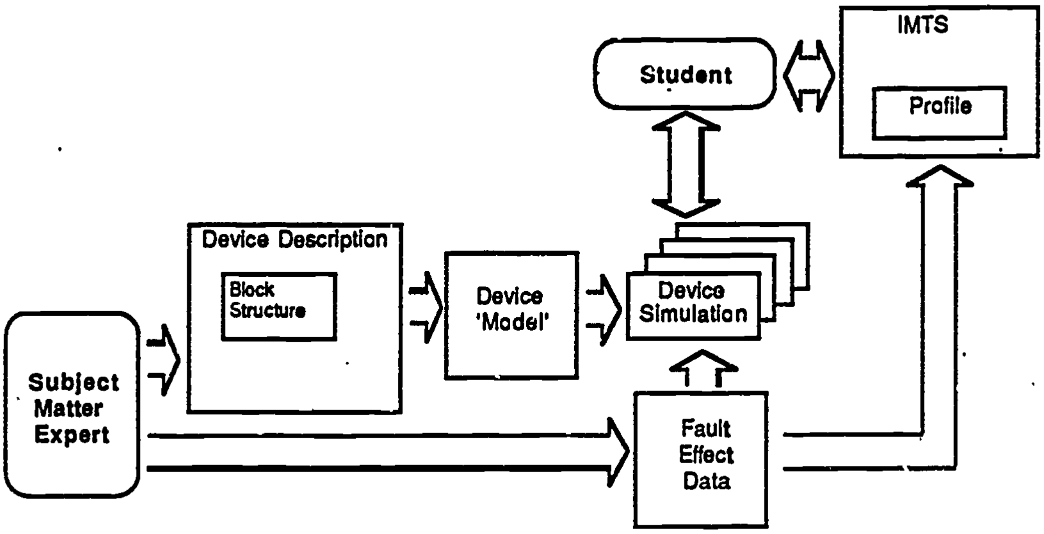
Simulation of Partially Specifiable Systems. Some equipment systems, such as Bladefold, lend themselves to simulation based on detailed descriptions at the component level. This level of description is appropriate when the number of low-level components (such as relay contact sets, switches, lights, and pressure snubbers) is not so large that it is infeasible to draw them, describe their behaviors, and specify their connections. This is the way that the IMTS Bladefold simulation has been implemented.

Other equipment systems are much too large and complex to be described in a similar manner, as too much time would be required to produce detailed behavior descriptions of every component. An alternative approach, capable of providing training for devices of arbitrary complexity, is required. For such complex systems, IMTS will make use of system level descriptions instead of component level descriptions.

In both approaches to IMTS simulation training, the aiding and instructional features are based on the Profile system's analysis of fault effect data. In the case of simulations based on detailed component descriptions, this fault effect information is *generated* automatically from the 'deep' device model. In the case of simulations based on system-level descriptions, a subject matter expert generates the fault effect data. For the latter type of IMTS simulations, the fault effect data are used to drive the simulation of the target equipment. For the former type, the simulation is generated by a device model based on detailed descriptions of the behavior of the device's components (see Figure 23).



IMTS Based on Detailed Component Descriptions



IMTS Based on System Level Descriptions

Figure 23. Two IMTS Approaches to Device Representation & Training

Authored Procedure Training. Instructors will be able to create guided simulations for teaching particular procedures. The authoring process will consist of putting a graphical simulation into *record* mode, and then carrying out the set of actions that they want to require their students to perform. Authors will also have the option of entering small blocks of text that can appear at specified points in the sequence of actions they record.

During the procedure training, students see the text blocks, which describe what is required of them. When a student performs a required action, the simulation updates and the next text block is presented. If the student performs some action other than the scripted (recorded) one, then the required action will be graphically highlighted.

The same methods will be applicable to other training requirements in addition to procedure training. For example, *ad hoc* instructional elements will be easily authored using the simulation record mode. Instructors who wish to teach a particular approach to troubleshooting a certain fault will be able to record the sequence of actions they use to troubleshoot the fault, adding explanatory text as they go. In the simulation playback mode, students will be guided to perform the same troubleshooting steps.

References

Anderson, J. R. *The architecture of cognition.* Cambridge, MA: Harvard University Press, 1983.

Anderson, J. R., Farrell, R., and Sauers, R. Learning to program in LISP. *Cognitive Science*, 1984, **8**, 87-130.

Beard, M. Barr A. V., Gould, L., and Wescourt, K. *Curriculum information networks for computer-assisted instruction* (Technial Report NPRDC TR 78-18). Stanford, CA: Institute for Mathematical Studies in the Social Sciences, Stanford University, 1978.

Bilodeau, I. McD. Information feedback. In E. A. Bilodeau (Ed.), *Principles of skill acquisiton.* New York: Academic Press, 1969.

Burton, R. R. and Brown, J. S. An investigation of computer coaching for informal learning activities. In D. Sleman and J. S. Brown (Eds.) *Intelligent tutoring systems.* London: Academic Press, 1982.

Hollan, J. D. 1983. STEAMER: An Overview with Implications for AI Applications in Other Domains. Presented at the Joint Services Workshop on Artificial Intelligence in Maintenance, Institute of Cognitive Science, Boulder, CO: October 4-6, 1983.

Hollan, J. D., Hutchins, E. L., and Weitzman, L. 1984. STEAMER: An Interactive Inspectable Simulation-based Training System, *The AI Magazine*, 1984, 2.

Johnson, M .C., Munro, A., and Towne, D. M. *Evaluation of the BTL Trainer in a Navy school environment.* (Technical Report No. 94) Los Angeles: University of Southern California, Behavioral Technology Laboratories, 1981.

McKendree, J., Reiser, B. J., and Anderson, J. R. Tutorial goals and strategies in the instruction of programming skills. Proceedings of the sixth annual conference of the cognitive science society. Boulder, CO: Institute for Cognitive Science, 1984.

Munro, A., Fehling, M. R., Blais, P., and Towne, D. M. Intrusive and non-intrusive instruction in dynamic skill training. (Technical Report No. 97)

Los Angeles: University of Southern California, Behavioral Technology Laboratories, 1981.

Munro, A., Towne, D. M., and King, C. A. A stand-alone generalized trainer-simulator. *Association for the Development of Computer-based Instructional Systems News*, 1980, 12, No. 4, 9 - 13.

Munro, A., Towne, D. M., and Surmon, D. S. *Representing dynamic skill knowledge*. (Technical Report No. 103) Los Angeles: Behavioral Technology Laboratories, University of Southern California, December 1984.

Norman, D. A. and Draper S. W. (Eds.) *User-centered system design: New perspectives on human-computer interaction..* Hillsdale, NJ: Lawrence Erlbaum Associates, 1986.

Skinner, B. F. *The technology of teaching*. New York: Appleton-Century-Crofts, 1968.

Towne, D. M. A generalized model of fault-isolation performance. *Proceedings, Artificial Intelligence in Maintenance: Joint Services Workshop*, 1984.

Towne, D.M. Cognitive Workload in Fault Diagnosis. Technical Report No 107, Los Angeles, CA: Behavioral Technology Laboratories, University of Southern California. 1985.

Towne, D. M. A generic expert diagnostician. In *The Proceedings of the Air Force Workshop on Artificial Intelligence Applications for Integrated Diagnostics*, 1986.

Towne, D. M. The generalized maintenance trainer: Evolution and revolution. In W. B. Rouse (Ed.), *Advances in man-machine systems research, Vol 3*, JAI Press, 1986.

Towne, D. M., Fehling, M. R., and Bond, N. A. *Design for the Maintainer: Projecting maintenance performance from design characteristics.* (Technical Report No. 95) Los Angeles, CA: Behavioral Technology Laboratories, University of Southern California, 1981.

Towne, D. M. and Johnson, M. C. *Computer Aiding of Maintainability Design: A Feasibility Study.* (Technical Report No. 104) Los Angeles, CA: Behavioral Technology Laboratories, University of Southern California, 1984

Towne, D. M. and Johnson, M. C. *Research on Computer Aided Design for Maintainability.* (Technical Report No. 109) Los Angeles, CA: Behavioral Technology Laboratories, University of Southern California, 1987

Towne, D. M., Johnson, M. C. and Corwin, W. H.. *A Technique for Projecting Maintenance Performance from Design Characteristics.* Technical Report No. 100. Los Angeles, CA: Behavioral Technology Laboratories, University of Southern California, 1982.

Towne, D. M., Johnson, M. C. and Corwin, W. H. *A Performance-based Technique for Assessing Equipment Maintainability.* (Technical Report No. 102) Los Angeles, CA: Behavioral Technology Laboratories, University of Southern California, 1983.

Towne, D. M. and Munro, A. *Generalized maintenance trainer simulator: Development of hardware and software.* (Technical Report No. 81-9) San Diego: Navy Personnel Research and Development Center, 1981.

Towne, D. M. and Munro, A. *Preliminary design of the advanced ESAS System.* (Technical Report No. 105) Los Angeles: Behavioral Technology Laboratories, University of Southern California, December 1984.

Towne, D. M., Munro, A., Johnson, M. C., and Lahey, G. F. *Generalized Maintenance Trainer Simulator: Test and Evaluation in the Laboratory Environment.* (NPRDC TR 83-28) San Diego: Navy Personnel Research and Development Center, August 1983.

Towne, D. M., Munro, A., Pizzini, Q. A., and Surmon, D. S. *Development of intelligent maintenance training technology: Design study.* (Technical Report No. 106) Los Angeles: Behavioral Technology Laboratories, University of Southern California, May 1985.

Towne, D. M., Munro, A., Pizzini, Q. A., and Surmon, D. S. *Representing System Behaviors and Expert Behaviors for Intelligent Tutoring.* Technical Report No. 108, Los Angeles: Behavioral Technology Laboratories, University of Southern California, 1987.

Towne, D. M., Pizzini, Q. A., Surmon, D. S., and Munro, A. *The Intelligent Maintenance Training System User's Manual*. Los Angeles: Behavioral Technology Laboratories,University of Southern California, 1987.

Tulving, E. *Elements of episodic memory*. London: Oxford University Press, 1983.

Tulving, E. and Thompson, D. M. Encoding specificity and retrieval processes in episodic memory. *Psychological Review*, 1973, 80, 352-373.

Wescourt, K. T., Beard, M., and Gould, L. Knowledge-based adaptive curriculum sequencing for CAI: Application of a network representation. *Proceedings of the Association for Computing Machinery*, 1977, 234-340.

Wescourt, K. T. and Hemphill, L. *Representing and teaching knowledge for troubleshooting/ debugging* (Technical Report 292). Stanford, CA: Institute for mathematical Studies in the Social Sciences, Stanford University, 1978.

Wescourt, K. T. , Beard, M., Gould, L., and Barr, A. *Knowledge-based CAI: CINS for individualized curriculum sequencing* (Technical Report 290). Stanford, CA: Institute for mathematical Studies in the Social Sciences, Stanford University, 1977.

ONR Distribution List (D. Towne) 4/88

Dr. Beth Adelson
Department of Computer Science
Tufts University
Medford, MA 02155

Personnel Analysis Division,
AF/MPXA
5C360, The Pentagon
Washington, DC 20330

Air Force Human
Resources Lab
AFHRL/MPD
Brooks, AFB, TX 78235

AFOSR,
Life Sciences Directorate
Bolling Air Force Base
Washington, DC 20332

Technical Director,
Army Human Engineering Lab
ATTN: SLCHE-D
Aberdeen Proving Ground
MD 21005-5001

Dr. Robert Ahlers
Code N711
Human Factors Laboratory
Naval Training Systems Center
Orlando, FL 32813

Dr. John R. Anderson
Department of Psychology
Carnegie-Mellon University
Schenley Park
Pittsburgh, PA 15213

Dr. Stephen J. Andriole, Chairman
Department of Information Systems
and Systems Engineering
George Mason University
4400 University Drive
Fairfax, VA 22030

Technical Director, ARI
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. Patricia Baggett
School of Education
610 E. University, Rm 1302D
University of Michigan
Ann Arbor, MI 48109-1259

Dr. Eva L. Baker
UCLA Center for the Study
of Evaluation
145 Moore Hall
University of California
Los Angeles, CA 90024

Dr. Meryl S. Baker
Navy Personnel R&D Center
San Diego, CA 92152-6800

prof. dott. Bruno G. Bara
Unita di ricerca di
intelligenza artificiale
Universita di Milano
20122 Milano - via F. Sfor3
ITALY

Dr. Gautam Biswas
Department of Computer Scie
Box 1688, Station B
Vanderbilt University
Nashville, TN 37235

Dr. John Black
Teachers College, Box 8
Columbia University
525 West 120th Street
New York, NY 10027

Dr. Deborah A. Boehm-Davis
Department of Psychology
George Mason University
4400 University Drive
Fairfax, VA 22030

Dr. Jeff Bonar
Learning R&D Center
University of Pittsburgh
Pittsburgh, PA 15260

Dr. Robert Breaux
Code 7B
Naval Training Systems Cenr
Orlando, FL 32813-7100

ONR Distribution List (D. Towne) 4/88

Dr. John S. Brown
XEROX Palo Alto Research
Center
3333 Coyote Road
Palo Alto, CA 94304

Dr. John T. Bruer
James S. McDonnell Foundation
Suite I610
1034 South Brentwood Blvd.
St. Louis, MO 63117

Dr. Bruce Buchanan
Computer Science Department
Learning R&D Center
University of Pittsburgh
3939 O'Hara St.
Pittsburgh, PA 15260

Dr. Raymond E. Christal
UES LAMP Science Advisor
AFHRL/MOEL
Brooks AFB, TX 78235

Professor Chu Tien-Chen
Mathematics Department
National Taiwan University
Taipei, TAIWAN

Dr. William Clancey
Institute for Research
on Learning
3333 Coyote Hill Road
Palo Alto, CA 94304

Assistant Chief of Staff
for Research, Developme,
Test, and Evaluation
Naval Education and
Training Command (N-5)
NAS Pensacola, FL 32508

Brian Dallman
Training Technology Branch
3400 TCHCTW/TTGXC
Lowry AFB, CO 80230-5000

Dr. Robert B. Davis
Curriculum Laboratory
(Education)
University of illinois
Urbana, IL 61801

Dr. Andrea di Sessa
University of California
School of Education
Tolman Hall
Berkeley, CA 94720

Defense Technical
Information Center
Cameron Station, Bldg 5
Alexandria, VA 22314
Attn: TC
(12 Copies)

ERIC Facility-Acquisitions
4350 East-West Hwy., Suite 1100
Bethesda, MD 20814-4475

Dr. Marshall J. Farr, Consultant
Cognitive & Instructional Sciences
2520 North Vernon Street
Arlington, VA 22207

Dr. Gerhard Fischer
University of Colorado
Department of Computer Science
Boulder, CO 80309

Dr. Kenneth D. Forbus
University of Illinois
Department of Computer Science
1304 West Springfield Avenue
Urbana, IL 61801

Dr. Barbara A. Fox
University of Colorado
Department of Linguistics
Boulder, CO 80309

Dr. John R. Frederiksen
BBN Laboratories
10 Moulton Street
Cambridge, MA 02238

Dr. Michael Friendly
Psychology Department
York University
Toronto ONT
CANADA M3J 1P3

Julie A. Gadsden
Information Technology
Applications Division
Admiralty Research Establishment
Portsdown, Portsmouth PO6 4AA
UNITED KINGDOM

Dr. Dedre Gentner
University of Illinois
Department of Psychology
603 E. Daniel St.
Champaign, IL 61820

Dr. Robert Glaser
Learning Research
& Development Center
University of Pittsburgh
3939 O'Hara Street
Pittsburgh, PA 15260

Dr. Sam Glucksberg
Department of Psychology
Princeton University
Princeton, NJ 08540

Dr. Sherrie Gott
AFHRL/MOMJ
Brooks AFB, TX 78235-5601

Dr. T. Govindaraj
Georgia Institute of
Technology
School of Industrial
and Systems Engineering
Atlanta, GA 30332-0205

Dr. Jordan Grafman
Neuropsychology Section
Medical Neurology
Branch-NINCDS
Building 10, Room 5C416
Bethesda, MD 20892

Dr. James G. Greeno
School of Education
Stanford University
Room 311
Stanford, CA 94305

Dr. Dik Gregory
Admiralty Research
Establishment/AXB
Queens Road
Teddington
Middlesex, ENGLAND TW110LN

Dr. Henry M. Halff
Halff Resources, Inc.
4918 33rd Road, North
Arlington, VA 22207

ONR Distribution List (D. Towne) 4/88

Dr. Bruce W. Hamill
Research Center
The Johns Hopkins University
Applied Physics Laboratory
Johns Hopkins Road
Laurel, MD 20707

Dr. Barbara Hayes-Roth
Knowledge Systems Laboratory
Stanford University
701 Welch Road
Palo Alto, CA 94304

Dr. James Hollan
NPRDC, UCSD
Code 501
San Diego, CA 92152

Dr. Keith Holyoak
Department of Psychology
University of California
Los Angeles, CA 90024

Ms. Julia S. Hough
Lawrence Erlbaum Associates
110 W. Harvey Street
Philadelphia, PA 19144

Dr. Ed Hutchins
Intelligent Systems Group
Institute for
Cognitive Science (C-015)
UCSD
La Jolla, CA 92093

Dr. Janet Jackson
Rijksuniversiteit Groningen
Biologisch Centrum, Vleugel D
Kerklaan 30, 9751 NN Haren
The NETHERLANDS

Dr. Marcel Just
Carnegie-Mellon University
Department of Psychology
Schenley Park
Pittsburgh, PA 15213

Dr. Daniel Kahneman
Department of Psychology
University of California
Berkeley, CA 94720

Dr. Milton S. Katz
European Science Coordinatn
Office
U.S. Army Research Institue
Box 65
FPO New York 09510-1500

Dr. Wendy Kellogg
IBM T. J. Watson Research .
P.O. Box 704
Yorktown Heights, NY 1058

Dr. David Kieras
Technical Communication Prm
TIDAL Bldg., 2360 Bonistee.
University of Michigan
Ann Arbor, MI 48109-2108

Dr. Janet L. Kolodner
Georgia Institute of Techny
School of Information
& Computer Science
Atlanta, GA 30332

Dr. Kenneth Kotovsky
Community College of
Allegheny County
808 Ridge Avenue
Pittsburgh, PA 15212

ONR Distribution List (D. Towne) 4/88

Dr. Claude Janvier
Universite' du Quebec a Montreal
P.O. Box 8888, succ: A"
Montreal, Quebec H3C 3P8
CANADA

Dr. Robin Jeffries
Hewlett-Packard Laboratories, 3L
P.O. Box 10490
Palo Alto, CA 94303-0971

Dr. Robert W. Lawler
Matthews 118
Purdue University
West Lafayette, IN 47907

Dr. Alan M. Lesgold
Learning R&D Center
University of Pittsburgh
Pittsburgh, PA 15260

Dr. Jim Levin
Department of
Educational Psychology
210 Education Building
1310 South Sixth Street
Champaign, IL 61820-6990

Dr. John Levine
Learning R&D Center
University of Pittsburgh
Pittsburgh, PA 15260

Dr. Clayton Lewis
University of Colorado
Department of Computer Science
Campus Box 430
Boulder, CO 80309

Library
Naval Training Systems Center
Orlando, FL 32813

Library
Naval War College
Newport, RI 02940

Dr. Benjamin Kuipers
University of Texas at Ausn
Department of Computer Scis
Taylor Hall 2.124
Austin, Texas 78712

Dr. Jill Larkin
Carnegie-Mellon University
Department of Psychology
Pittsburgh, PA 15213

Dr. William L. Maloy
Naval Education and Trainig
Program Support Activity
Code 047
Building 2435
Pensacola, FL 32509-5000

Dr. Elaine Marsh
Naval Center for Applied Rh
in Artificial Intelligene
Naval Research Laboratory
Code 5510
Washington, DC 20375-5000

Dr. Sandra P. Marshall
Dept. of Psychology
San Diego State University
San Diego, CA 92182

Dr. Manton M. Matthews
Department of Computer Scie
University of South Carolia
Columbia, SC 29208

Dr. Joseph C. McLachlan
Code 52
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr. James McMichael
Technical Director
Navy Personnel R&D Center
San Diego, CA 92152-6800

ONR Distribution List (D. Towne) 4/88

Science and Technology Division
Library of Congress
Washington, DC 20540

Vern M. Malec
NPRDC, Code 52
San Diego, CA 92152-6800

Dr. Jane Malin
Mail Code EF5
NASA Johnson Space Center
Houston, TX 77058

Dr. Allen Munro
Behavioral Technology
Laboratories - USC
1845 S. Elena Ave., 4th Floor
Redondo Beach, CA 90277

Spec. Asst. for Research, Experimental & Academic Programs,
NTTC (Code 016)
NAS Memphis (75)
Millington, TN 38054

Dr. Donald A. Norman
C-015
Institute for Cognitive Science
University of California
La Jolla, CA 92093

Deputy Technical Director
NPRDC Code 01A
San Diego, CA 92152-6800

Dr. Barbara Means
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025

Dr. James R. Miller
MCC
3500 W. Balcones Center Dr.
Austin, TX 78759

Dr. William Montague
NPRDC Code 13
San Diego, CA 92152-6800

Dr. Randy Mumaw
Training Research Division
HumRRO
1100 S. Washington
Alexandria, VA 22314

Dr. Harold F. O'Neil, Jr.
School of Education - WPH 1
Department of Educational
Psychology & Technology
University of Southern Cal.
Los Angeles, CA 90089-001

Office of Naval Research,
Code 1142
800 N. Quincy St.
Arlington, VA 22217-5000

Office of Naval Research,
Code 1142BI
800 N. Quincy Street
Arlington, VA 22217-5000

Office of Naval Research,
Code 1142CS
800 N. Quincy Street
Arlington, VA 22217-5000
(6 Copies)

ONR Distribution List (D. Towne) 4/88

Director, Training Laboratory,
NPRDC (Code 05)
San Diego, CA 92152-6800

Director, Manpower and Personnel
Laboratory,
NPRDC (Code 06)
San Diego, CA 92152-6800

Director, Human Factors
& Organizational Systems Lab,
NPRDC (Code 07)
San Diego, CA 92152-6800

Library, NPRDC
Code P201L
San Diego, CA 92152-6800

Technical Director
Navy Personnel R&D Center
San Diego, CA 92152-6800

Commanding Officer,
Naval Research Laboratory
Code 2627
Washington, DC 20390

Military Assistant for Training and
Personnel Technology,
OUSD (R & E)
Room 3D129, The Pentagon
Washington, DC 20301-3080

Office of Naval Research,
Code 1142PS
800 N. Quincy Street
Arlington, VA 22217-5000

Psychologist
Office of Naval Research
Branch Office, London
Box 39
FPO New York, NY 09510

Special Assistant for Marine
Corps Matters,
ONR Code 00MC
800 N. Quincy St.
Arlington, VA 22217-5000

Dr. Judith Orasanu
Basic Research Office
Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. Douglas Pearce
1133 Sheppard W
Box 2000
Downsview, Ontario
CANADA M3M 3B9

Dr. James A. Reggia
University of Maryland
School of Medicine
Department of Neurology
22 South Greene Street
Baltimore, MD 21201

ONR Distribution List (D. Towne) 4/88

Dr. Tjeerd Plomp
Twente University of Technology
Department of Education
P.O. Box 217
7500 AE ENSCHEDE
THE NETHERLANDS

Dr. Martha Polson
Department of Psychology
University of Colorado
Boulder, CO 80309-0345

Dr. Steven E. Poltrock
MCC
3500 West Balcones Center Dr.
Austin, TX 78759-6509

Dr. Harry E. Pople
University of Pittsburgh
Decision Systems Laboratory
1360 Scaife Hall
Pittsburgh, PA 15261

Dr. Joseph Psotka
ATTN: PERI-IC
Army Research Institute
5001 Eisenhower Ave.
Alexandria, VA 22333-5600

Mr. Paul S. Rau
Code U-33
Naval Surface Weapons Center
White Oak Laboratory
Silver Spring, MD 20903

Dr. Steve Reder
Northwest Regional
Educational Laboratory
400 Lindsay Bldg.
710 S.W. Second Ave.
Portland, OR 97204

Dr. J. Wesley Regian
AFHRL/IDI
Brooks AFB, TX 78235

Dr. Fred Reif
Physics Department
University of California
Berkeley, CA 94720

Dr. Brian Reiser
Department of Psychology
Green Hall
Princeton University
Princeton, NJ 08540

Dr. Gilbert [illegible]card
Mail Stop K02-14
Grumman Aircraft Systems
Bethpage, NY 11787

Dr. J. Jeffrey Richardson
Center for Applied AI
College of Business
University of Colorado
Boulder, CO 80309-0419

Dr. Linda G. Roberts
Science, Education, and
Transportation Program
Office of Technology Assest
Congress of the United States
Washington, DC 20510

Dr. William B. Rouse
Search Technology, Inc.
725 Peachtree Corners Circ
Suite 200
Norcross, GA 30092

Dr. Walter Schneider
Learning R&D Center
University of Pittsburgh
3939 O'Hara Street
Pittsburgh, PA 15260

ONR Distribution List (D. Towne) 4/88

Dr. Janet W. Schofield
816 LRDC Building
University of Pittsburgh
3939 O'Hara Street
Pittsburgh, PA 15260

Dr. Miriam Schustack
Code 52
Navy Personnel R & D Center
San Diego, CA 92152-6800

Dr. Judith W. Segal
OERI
555 New Jersey Ave., NW
Washington, DC 20208

Dr. Colleen M. Seifert
Institute for Cognitive Science
Mail Code C-015
University of California, San Diego
La Jolla, CA 92093

Dr. Ben Shneiderman
Dept. of Computer Science
University of Maryland
College Park, MD 20742

Dr. Randall Shumaker
Naval Research Laboratory
Code 5510
4555 Overlook Avenue, S.W.
Washington, DC 20375-5000

Dr. Edward E. Smith
Department of Psychology
University of Michigan
330 Packard Road
Ann Arbor, MI 48103

Dr. Kurt Steuck
AFHRL/MOA
Brooks AFB
San Antonio, TX 78235-5601

Dr. Albert Stevens
Bolt Beranek & Newman, Inc.
10 Moulton St.
Cambridge, MA 02238

Dr. John Tangney
AFOSR/NL, Bldg. 410
Bolling AFB, DC 20332-6448

Dr. Kikumi Tatsuoka
CERL
252 Engineering Research
Laboratory
103 S. Mathews Avenue
Urbana, IL 61801

Dr. Perry W. Thorndyke
FMC Corporation
Central Engineering Labs
1205 Coleman Avenue, Box 50
Santa Clara, CA 95052

Dr. Martin A. Tolcott
3001 Veazey Terr., N.W.
Apt. 1617
Washington, DC 20008

Dr. Douglas Towne
Behavioral Technology Labs
University of Southern Cala
1845 S. Elena Ave.
Redondo Beach, CA 90277

ONR Distribution List (D. Towne) 4/88

Dr. Elliot Soloway
Yale University
Computer Science Department
P.O. Box 2158
New Haven, CT 06520

Dr. Richard C. Sorensen
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr. Kathryn T. Spoehr
Brown University
Department of Psychology
Providence, RI 02912

Dr. Keith T. Wescourt
FMC Corporation
Central Engineering Labs
1205 Coleman Ave., Box 580
Santa Clara, CA 95052

Dr. Kent E. Williams
Inst. for Simulation and Training
University of Central Florida
P.O. Box 25000
Orlando, FL 32816-0544

Dr. Martin F. Wiskoff
Defense Manpower Data Center
550 Camino El Estero
Suite 200
Monterey, CA 93943-3231

Dr. Masoud Yazdani
Dept. of Computer Science
University of Exeter
Prince of Wales Road
Exeter.EX44PT
ENGLAND

Headquarters, U. S. Marines
Code MPI-20
Washington, DC 20380

Dr. Jerry Vogt
Navy Personnel R&D Center
Code 51
San Diego, CA 92152-6800

Dr. Ralph Wachter
JHU-APL
Johns Hopkins Road
Laurel, MD 20707

Dr. Barbara White
BBN Laboratories
10 Moulton Street
Cambridge, MA 02238

Dr. Robert A. Wisher
U.S. Army Institute for the
Behavioral and Social Sciences
5001 Eisenhower Avenue
Alexandria, VA 22333-5600

Dr. Wallace Wulfeck, III
Navy Personnel R&D Center
Code 51
San Diego, CA 92152-6800

Dr. Joseph L. Young
National Science Foundation
Room 320
1800 G Street, N.W.
Washington, DC 20550

Dr. Hillel Einhorn
Graduate School of Business
University of Chicago
1101 E. 58th Street
Chicago, IL 60637

Dr. Douglas Towne
University of Southern California
Behavioral Technology labs
1845 S. Elena Ave., 4th Flr.
Redondo Beach, CA 90277

Dr. John Payne
Graduate School of Business Administration
Duke University
Durham, NC 27706

Dr. Dana Yoerger
Deep Submergence Laboratory
Woods Hole Ocenaographic Institution
Woods Hole, MA 02543

Paul Christopher
University of Southern California
University Library, mc-0183
University Park
Los Angeles, CA 90089-0183

Dr. Paul Slovic
Decision Research
1201 Oak Street
Eugene, OR 97401

Dr. Nicholas Bond
4701 Anne Way
Carmichael, CA 95608

Dr. Jesse Orlansky
Institute of Defense Analyses
1801 N. Beauregard Street
Alexandria, VA 22311

Dr. Azad Madni
Perceptronics, Inc.
6271 Variel Ave.
Woodland Hills, CA 91364

Dr. Anthony Mason
CAL POLY
Department of Industrial Eng.
San Luis Obispo, CA 93407