N96-14988

Advanced Technology Training System on Motor-Operated Valves

Bradley J. Wiederholt
T. Kiki Widjaja
Galaxy Scientific Corporation
Atlanta, GA

Joseph Y. Yasutake
Electric Power Research Institute
Palo Alto, California

Hachiro Isoda
Central Research Institute of Electric Power Industry
Tokyo, Japan

This paper describes how features from the field of Intelligent Tutoring Systems are applied to the Motor-Operated Valve (MOV) Advanced Technology Training System (ATTS). The MOV ATTS is a training system developed at Galaxy Scientific Corporation for the Central Research Institute of Electric Power Industry in Japan and the Electric Power Research Institute in the United States. The MOV ATTS combines traditional computer-based training approaches with system simulation, integrated expert systems, and student and expert modeling.

The primary goal of the MOV ATTS is to reduce human errors that occur during MOV overhaul and repair. The MOV ATTS addresses this goal by providing basic operational information of the MOV, simulating MOV operation, providing troubleshooting practice of MOV failures, and tailoring this training to the needs of each individual student.

The MOV ATTS integrates multiple expert models (functional and procedural) to provide advice and feedback to students. The integration also provides expert model validation support to developers. Student modeling is supported by two separate student models: one model registers and updates the student's current knowledge of basic MOV information, while another model logs the student's actions and errors during troubleshooting exercises. These two models are used to provide tailored feedback to the student during the MOV course.

INTRODUCTION

In February, 1989, the Electric Power Research Institute (EPRI) and the Central Research Institute of Electric Power Industry (CRIEPI) in Japan initiated a joint research program to investigate various interventions to reduce personnel errors and inefficiencies in the maintenance of nuclear power plants. One maintenance task identified as being particularly susceptible to human errors was the overhaul of Motor-Operated Valves (MOVs). MOVs are electro-mechanical devices used throughout nuclear power plants in many different systems, both safety-related and balance-of-plant. Because these valves are so numerous, systemic problems with any part of an MOV assembly can have negative effects on plant performance, including increases in (1) person-hours required for repair, (2) personnel radiation exposure, and (3) outage length (ANACAPA report).

A study of the MOV actuator overhaul task showed that MOV maintenance personnel must deal with a technically complex system, have limited access to the real equipment for either on-the-job or classroom training, and rely on experience (either their own or a group leader's) rather that written procedures for troubleshooting and diagnosis of the MOV actuator. In addition, requirements for the application of the personnel's troubleshooting knowledge are infrequent, requiring retraining or practice to maintain proficiency in the diagnostic-intensive parts of their job. The study also indicated that two separate occupational groups perform maintenance on MOV actuators: mechanics and electricians. Poor understanding of one another's skills and knowledge further increases the chance of on-the-job maintenance errors.

To train personnel to perform a task exhibiting the above characteristics, nuclear plant instructors currently augment stand-up classroom training with hands-on exercises. However, the lack of availability of both instructor time and real-world equipment limits the amount of effective hands-on training that instructors can provide. Informal interviews with instructors pointed to a need to provide students with training that could 1) increase student's exposure to MOV actuator troubleshooting tasks, 2) reduce the amount of subjectivity in evaluating a student's troubleshooting skills and knowledge, and 3) reduce the amount of instructor-led MOV retraining.

As a result of these needs and task analyses, CRIEPI and EPRI initiated the development of a computer-based training system to train nuclear power plant personnel in the diagnosis of MOV actuator problems. Initial review identified simulation-oriented tutoring as a training method most suited to help utility maintenance personnel gain a better understanding of MOV actuator operations and diagnosis (Widjaja analysis, 1992). For adult learners, various forms computer-based simulations of equipment for the purpose of teaching diagnostic skills have been repeatedly demonstrated to be effective within military and industrial environments. A previous EPRI project demonstrated the effectiveness of computer simulation to provide diagnostic training in the area of diesel generators (Johnson, et al., 1986, Maddox, et al., 1986).

Often these simulations are augmented with training and feedback that are tailored to the individual student's needs and understanding of the system. This adaptability of the training to the student is important to eliminate unnecessary training, focus students on their areas of weakness, and increase the acceptance of the training by the student. This simulation-oriented, "intelligent" feedback approach is particularly effective for teaching troubleshooting in systems that are 1) technically complex, 2) have varied instrumentation and test points, 3) require practice on the part of maintenance personnel to develop and maintain diagnostic proficiency, 4) have either formal or informal troubleshooting procedures, and 5) have limited access to the real equipment for training purposes (Norton, et al., 1991).

The MOV Advanced Technology Training System (ATTS) is the result of this development effort. The MOV ATTS provides basic operational information of the MOV, simulates MOV operation, provides 17 troubleshooting practice problems allowing student to diagnose MOV failures, and tailors this training to the needs of each individual student. The MOV ATTS is designed to be used by maintenance personnel of MOV actuator overhaul as a supplement to current initial training and refresher courses, and is not intended to replace these courses.

The remainder of this paper describes the internal features of the MOV ATTS, focusing on those aspects of the training that are based in the field of Intelligent Tutoring Systems.

GENERAL ARCHITECTURE

To support the goal of providing training that is tailored to the needs of each individual student, we have employed concepts and techniques from the field of Intelligent Tutoring Systems (ITS)[1] in the development of the MOV ATTS. A widely accepted representation of the architecture for an ITS is given in Figure 1 (Burns & Capps, 1988). The major components of an intelligent tutoring system are: 1) the *student*, 2) the computer *interface*, 3) the *instructional environment*, and 4) the *tutoring component*, which is in turn made up of models of the <u>student</u>, <u>expert</u> and <u>instructor</u>.

For purposes of the MOV ATTS, the computer interface consists simply of the physical interface (a computer monitor, mouse, keyboard and optional touch screen). The instructional environment, on the other hand, is the set of actions taken by the student and the feedback generated by the system during which the student performs the learning task. In the case of the MOV ATTS, there are two distinct instructional environments.

---

[1] General information on Intelligent Tutoring Systems can be found in Massey, et al., 1988 and Polsen and Richardson, 1988.

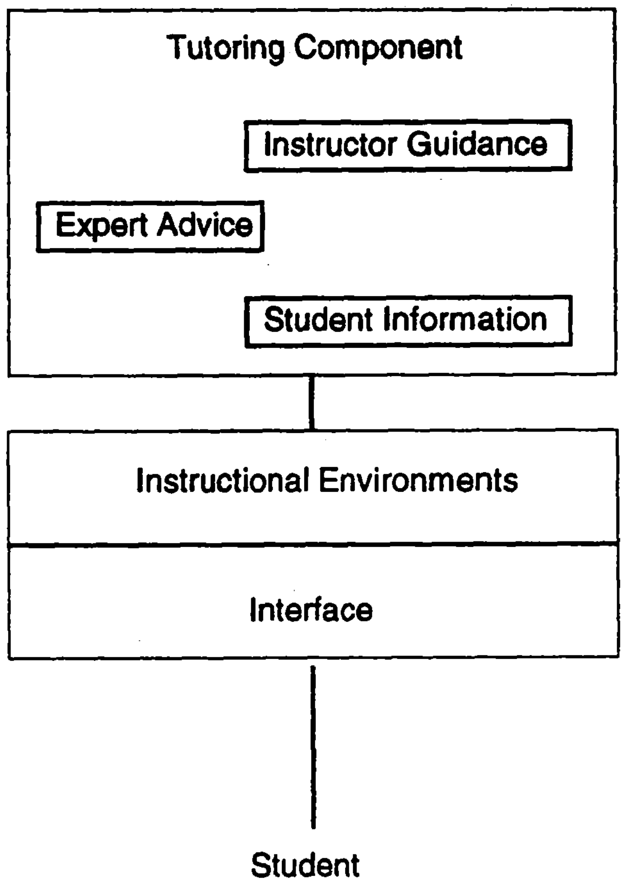


Figure 1. Simplified ITS Architecture.

In the first instructional environment, the student receives basic information on the MOV actuator, its mechanical components, and its electrical components. The student is occasionally called upon to answer question regarding various aspects of these topics. This self-paced material is presented to refresh the student's knowledge of the background material and to prepare them for the troubleshooting exercises to follow.

The MOV ATTS's second instructional environment is the system simulation (the troubleshooting module). Within the system simulation (see Figure 2), the student is actively involved in applying their knowledge to diagnose and locate failed components on a simulated MOV actuator. Within this environment, the student is testing components, receiving feedback when mistakes are made, and is trying to locate MOV failures as quickly as possible without making errors.

Note that according to Figure 1, both the basic information and system simulation instructional environments can be supported by the tutoring component. In fact, various portions of the MOV ATTS's tutoring component (the student, expert, and instructional models) are distributed among both instructional environments. The remainder of this paper will examine these components, where they are used in the MOV ATTS, how they interact with one another, and the benefits derived from their application.

STUDENT INFORMATION

In an ITS, the student information model is used to capture information on the student's current understanding of the subject matter being taught. The MOV ATTS actually uses two different student models: the declarative information model and the action history model.

The MOV ATTS declarative information student model contains 20 basic learning objectives similar to the example given above. These low level objectives were derived from an earlier analysis phase of the MOV ATTS project. Presently the declarative model is a linear representation of basic learning objectives. Extensions could be made in future applications of this approach to arrange them hierarchically, based on formal training needs analyses. Updating the higher order objectives would be based on propagating changes made to lower order objectives upward through the hierarchy, yielding a general approximation of the student's knowledge for the prescribed learning objectives.

Action History Student Model

The second student model attempts to create a history of actions and errors made by the student during troubleshooting exercises. Example actions that are logged to this action history model are troubleshooting tests performed, display screens seen, number and types of errors made, and time to solve a problem or perform an action. In short, all actions during a troubleshooting exercises are logged to the action history model. This action history is used by the expert and instructor models to provide advice as deemed appropriate. At the start of each new troubleshooting problem, the action history model from the previous problem is deleted and a new one is created.

The expert model uses the information contained in the action history model during the advice generation process. When the student solicits expert advice, the expert model will compare the suggested diagnostic procedure to the student's action history. When a discrepancy is found between what the student has done, and what the student should do, then the expert model uses this information to generate advice (see Expert Advice).

The instructor model uses the action history model to determine if the student is using the full power of the training system, or if the student is straying too far from common sense types of troubleshooting (e.g., making numerous redundant actions). More information on how this model is used in the Instructional Advice section.

This action history list is relatively simple when compared to the declarative information model, but yields sufficient information to support both the expert and instructional advice systems. The use of this type of student model dates back to 1985 for use in the diesel generator simulation project discussed earlier, and has evolved over the years through the Microcomputer Intelligence for Technical Training (MITT) projects performed for the National Aeronautics and Space Administration and the U.S. Air Force (Norton, et al., 1991).

EXPERT ADVICE

The expert model in an ITS is used to capture the expertise and knowledge about a domain that is to be transferred to the student. The information is captured and represented in such a way, though, as to support subsequent computation upon the information. This is in contrast to the way that conventional computer-based training systems capture, store, and process knowledge. Often these systems simply collect a set of facts, assign this material to a particular screen, and then present this static information to the student without any further processing.

The MOV ATTS contains two separate representations of expert knowledge: a rule-based procedural expert model, and a functional model of the system. Both of these models are used during the troubleshooting training exercises.

The Procedural Expert Model

The first expert model is a procedural model of how an effective troubleshooter would diagnose and repair an MOV actuator. This model is used to determine the correct diagnosis procedure for a given problem. As input, the procedural expert model reviews the current state of the simulated system, actions taken by the student so far, and specific real-world troubleshooting procedures. The procedural expert model then processes this information to generate the next step the student should take in the troubleshooting procedure (see Figure 3).

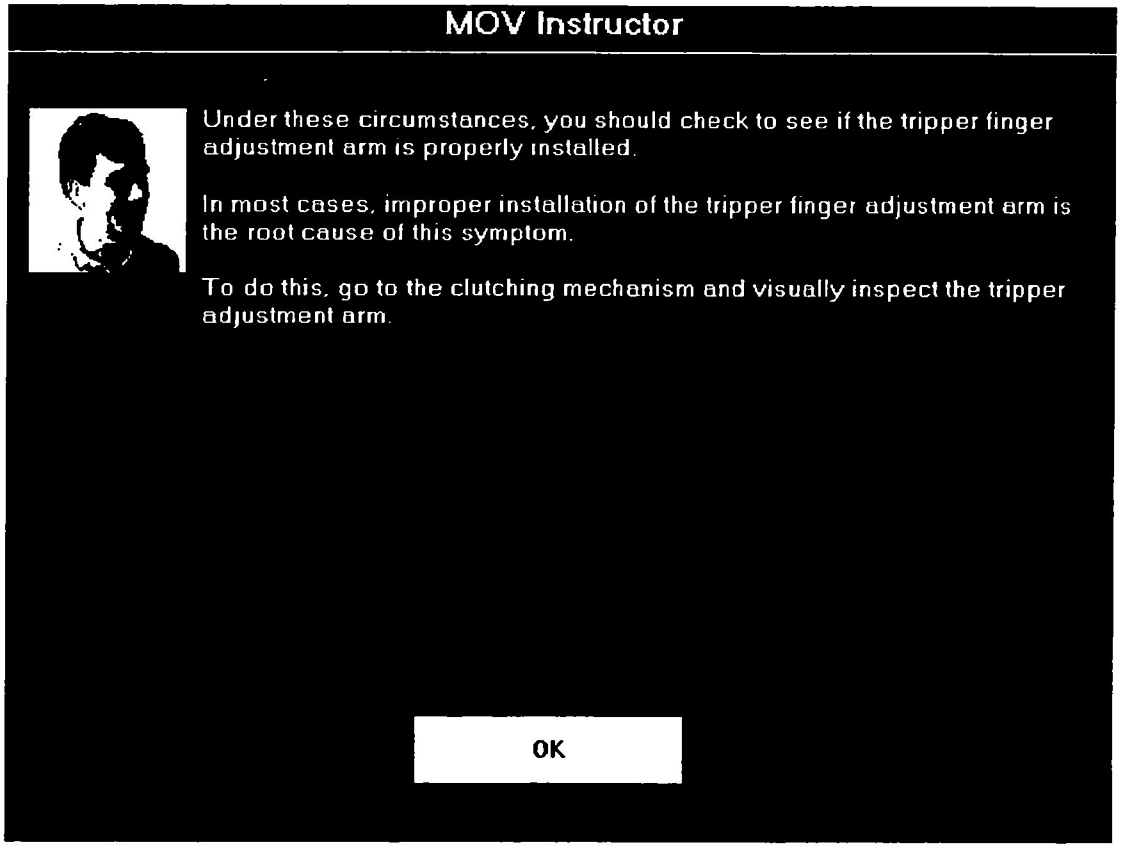


Figure 3. Sample Procedural Advice.

The real-world troubleshooting procedures used in the MOV ATTS were derived from informal interviews with MOV actuator subject-matter experts. The procedures were also based upon written rules and procedures in nuclear utility training manuals and the manufacturer's manuals.

The procedural expert model is put into action when a student requests advice during a troubleshooting exercise. The procedural expert first initializes itself with facts that it can gather about the system (e.g., facts about the state of components, the valve, indicator lights, etc., as well as information about what the student has done to this point). The procedural expert then scans its set of procedures to find the one that matches the initial conditions of the valve.

The procedural expert will then scan the set of actions and tests of the retrieved procedure, looking for an action that is appropriate at this stage in the troubleshooting process. This choice of an appropriate action is based on the results of various tests performed by the student and conditions of the simulated MOV actuator. Once an action is found, it is checked against the student's action history model to see if this action has already been performed. If it has not, it is presented to the student as the suggested next action to take. If the action has already been performed, the procedural expert will find the next appropriate action.

For example, an action that was suggested by the MOV instructors at Duke Power company was to always check the handwheel and declutch lever first when there is a problem switching the actuator from manual to remote. If there is still a problem even when these components are operating normally, then the problem is likely to involve either the tripper adjustment arm or the declutch fork. In the procedural expert, this knowledge is represented as:

```
if      problem-symptom is inability to switch from manual to remote and
        handwheel is feasible and
        clutching-subsystem is feasible
```

then

set advice to "Hold down the declutch lever and see if manual operation is possible in this position. To do this, click on the button labeled "Declutch Lever."

set explanation to "If manual operation is not possible even when you hold down the declutch lever then the problem is probably with the tripper adjustment arm or the declutch fork."

The procedural expert for the MOV ATTS contains approximately 75 of these procedural rules to cover the troubleshooting problems in the tutor. There is some economy of scale with the number of rules, i.e., as the number of problems increase, number of rules to be added for a new problem tend to decrease due to the existence of applicable rules from other procedures.

Functional Expert Model

The second expert model that is contained in the MOV ATTS is a functional model of the MOV actuator. This model is based upon individual parts in the system, and the influences that these parts have upon one another. This model evolved from previous experimental evaluations for support of diagnostic learning (Johnson, 1981; Rouse and Hunt, 1984) and have formed a core technology in many of our training systems since then.

Building an accurate functional expert model of a technical system requires detailed knowledge of the system, as well as a sharp understanding of the dependencies within this system. Even though our functional model of the MOV actuator (see Figure 4) only contains approximately 50 parts, it was a time consuming task to develop and refine this model and involved the participation of three Duke Power personnel and two Galaxy Scientific personnel.

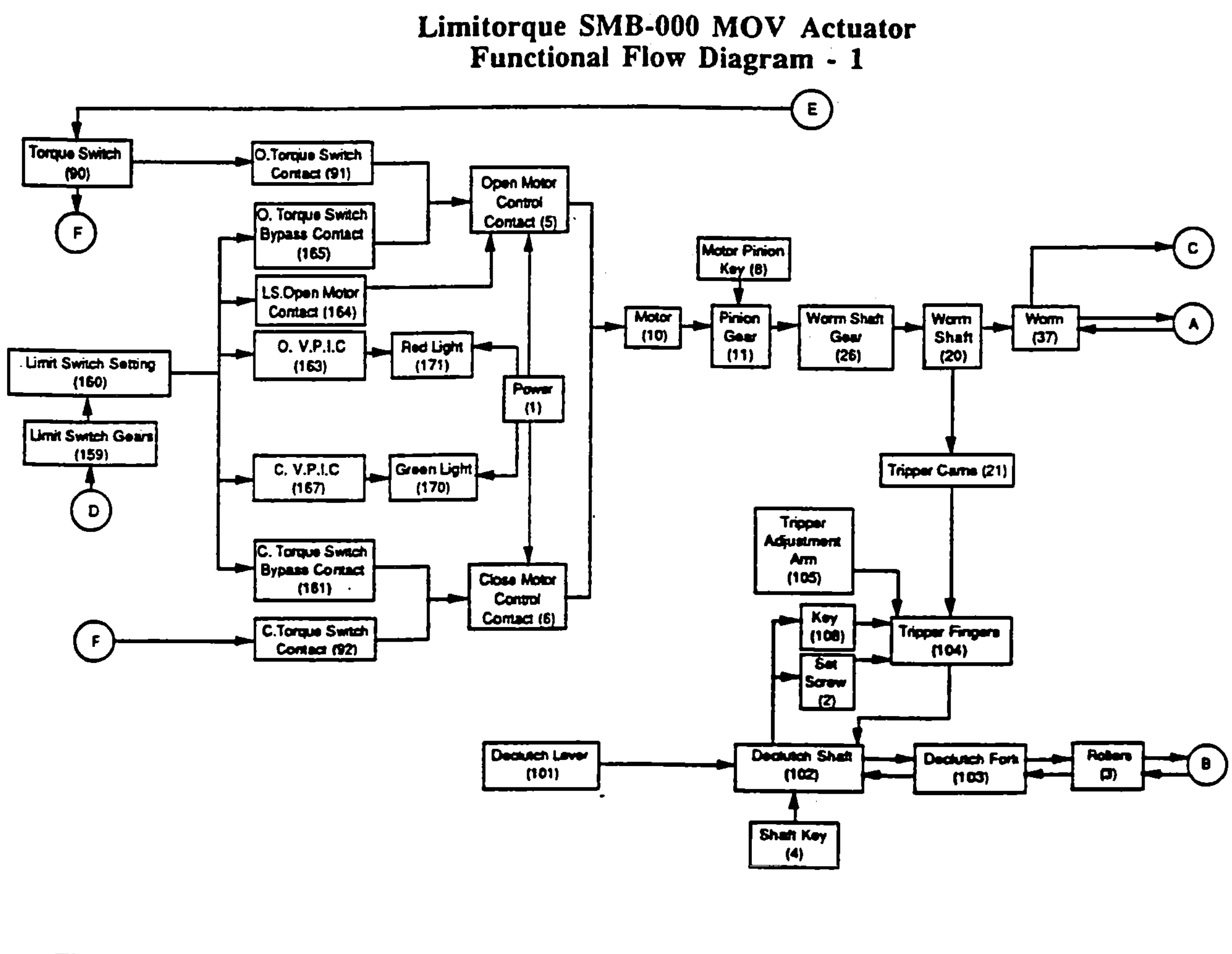


Figure 4. Sample section from MOV Functional Flow Diagram.

The functional expert model works on the concept of *functional dependency*. Functional dependency states that for a component in a system to function properly, all components that either directly or indirectly influence that part must also be functioning properly. For example, in Figure 4, in order for the Worm Shaft Gear (26) to function properly, this part and all parts that influence it must be functioning properly. This include the Pinion Gear (11), Motor Pinion Key, Motor (10), Open Motor Contact, Torque Switch (90), etc.

The MOV ATTS uses this functional expert model of the system in three ways. First, the functional expert model of the system is used to verify the procedural advice generated by the procedural expert model. Before the procedural expert presents advice to the student, it first checks the functional model to see if this piece of advice is logical. Note that the procedural expert is based upon human rules of thumb and advice, and is subject to contain some misleading advice or to overlook items that occur rarely in the real world. The functional model of the system, on the other hand, is completely logical in its representation of the system. By checking the procedural advice against this functional representation, we can avoid giving advice to the student based upon faulty expert logic (see Figure 5).

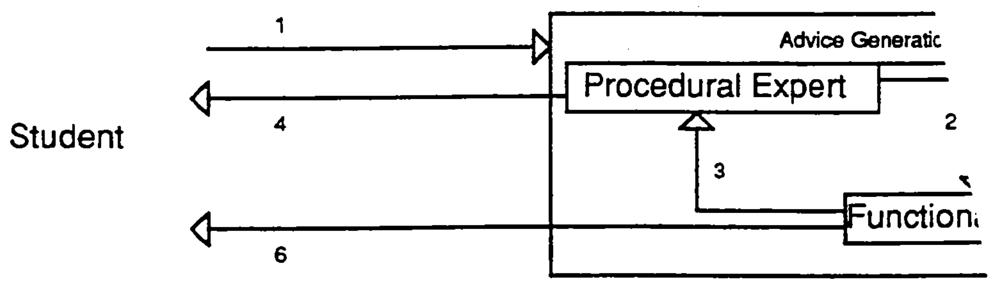


Figure 5. Advice generation by integrating functional and procedural models. *1) Student requests advice. 2) Procedural expert takes first attempt to generates advice. This is passed along to functional expert. 3) Functional expert either confirms or rejects advice. 4) If advice is confirmed, procedural expert formats advice and passes along to student. 5) If not confirmed, procedural advisor may generate more advice (via path 2) or may exhaust all procedural possibilities. When no more procedural advice is found, advice generation responsibilities are surrendered to functional expert via path 5. 6) Functional expert formats functional advice and passes along to student.*

One of the benefits of integrating both models is the ability to capture real-world shortcuts and procedures to teach the student (via the expert based procedural model), while maintaining the validity of these shortcuts (via the functional model of the system) before they become misconceptions. Even though there have been a few other ITSs that use dual expert models, the MOV ATTS represents a new step toward the integration of multiple experts in an ITS. In addition, the self-validation process provides a check-and-balance feature capable of aiding developers in the validation and debugging of rule-based expert models.

The second use of the functional expert model by the MOV ATTS is to provide a backup advice generator when the procedural expert cannot produce any further advice for a troubleshooting situation. The procedural advisor will only generate advice when it meets situations that it is programmed for. An example situation is when the procedural advisor has led the student to a point in which there are just a handful of remaining parts, and there is no predetermined procedures to further eliminate additional parts. The functional advisor is capable of taking over at this point and generating advice for the student.

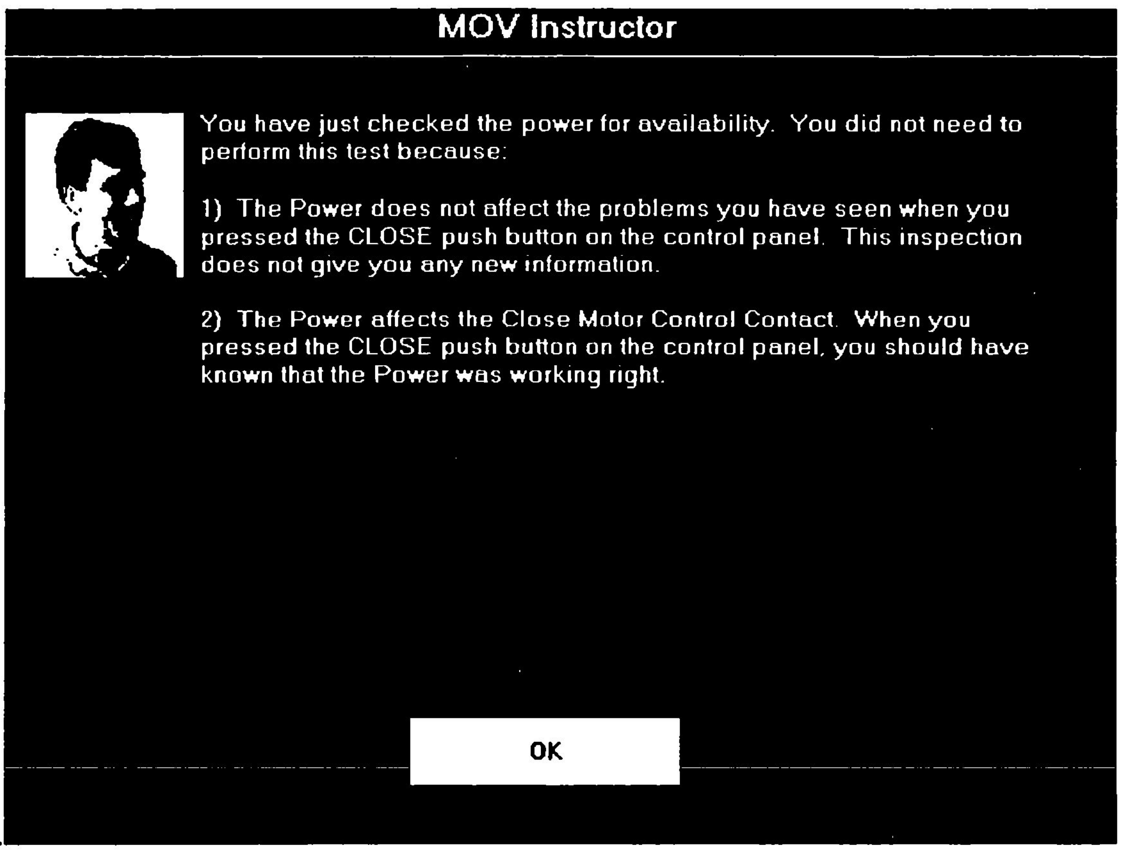
The functional advisor can generate advice at this point based upon the functional dependency of parts that have not been eliminated as a root cause of the problem. Generating this advice is accomplished basically by "half-splitting" the remaining parts, and determining the test or action that will eliminate the most parts from the feasible set. For example, if you had 4 parts in a system connected serially, the most logical test is to test the connection parts 2 and 3. If the results of the test are negative, you know that either part 1 or 2 has failed. If the results of the test are positive, you know that either part 3 or 4 is the failed part. The functional advisor is able to perform this half-splitting technique upon complex networks of parts, such as the MOV actuator representation, yielding logical advice that will produce the minimum number of tests to locate the failed component.

Note that if the system did not contain any procedural rules at all, then the functional advisor would produce advice based on this simple half-splitting technique. However, the procedural advice is important in order to capture and pass along troubleshooting techniques that are equipment and site specific, and that can offer a time and cost savings during the troubleshooting tasks.

The third manner in which the functional expert is used is to call the student's attention to simple diagnostic errors being made. This feedback manifests itself as unsolicited advice during a troubleshooting exercise (see Figure 6). As the student performs tests and actions in the troubleshooting exercise, a comparison is made between these actions and the functional expert. If a student's test is illogical according to the model, one of three types of responses are generated: 1) the test is declared incorrect because this test or a logically equivalent test was performed earlier, 2) the test is declared incorrect because the tested component functionally influences a component or subsystem that is known to be functioning properly from previous tests, 3) the test is declared incorrect because the tested component does not functionally influence a component or subsystem that is known to be functioning improperly.

Instructional Guidance

Instructional guidance is used in the MOV ATTS to select appropriate lessons for the student, and to guide the student through the troubleshooting exercises. The first use of the instructor model determines the amount of the basic MOV, mechanical, and electrical material that is to be presented to the student. Most of the work for this use is done by the declarative information student model, who has calculated an estimate of the student's understanding of the individual learning objectives.



on.

Figure 6. Unsolicited advice from the functional expert.

The instructor model at this point simply looks for learning objectives that exceed a predetermined threshold (in the case of the MOV ATTS, this threshold is 0.80) and exempts students from any modules related to these learning objectives. Note that the students are given the option to review the exempted modules if they so desire.

As mentioned earlier, the instructor model can also review the declarative information model and suggest to the student to review particular modules in which the student has demonstrated weak understanding. This is done at the end of each major module and before the student is allowed to practice troubleshooting. Note that the student, if he or she desires, can ignore the advice of the instructor and continue

The second use of the instructional guidance comes into effect during the troubleshooting exercises. Within this module, the function of instructional guidance is to call the student's attention to simple diagnostic errors being made. This guidance manifests itself as unsolicited advice during a troubleshooting exercise. The instructor model, using the action history student model, records the number of times advice is requested, the particular section of the MOV actuator being diagnosed, the number of redundant or unnecessary actions (as determined by the functional expert), and the number of diagnostic procedural errors (as determined by the procedural expert). The instructor model periodically checks these counts during a troubleshooting exercise and if they exceed a predetermined amount (e.g., if the student makes 3 unnecessary actions), the instructor model will offer unsolicited feedback to the student based upon these rules. The benefit of this approach is to direct the student's attention to mistakes at the time they are made, rather than waiting to the end of the troubleshooting exercise to recap and summarize these points.

Summary

The primary goal of the MOV Advanced Technology Training System (ATTS) is to reduce human errors that have occurred during MOV overhaul and repair. The MOV ATTS addresses this goal by providing basic operational information of the MOV, by simulating the MOV operation, and by providing troubleshooting practice on the diagnosis of MOV failures, and by tailoring this training to the needs of each individual student.

An pilot effectiveness study of the MOV ATTS was conducted at the Maintenance Training Facility of the Tennessee Valley Authority's Bellefonte Nuclear Station. Six mechanical and six electrical maintenance personnel participated in the study which was conducted at the site during November 3 - 6, 1992. All twelve subjects were first trained on the declarative knowledge portion of the MOV ATTS. The control group was then trained on troubleshooting techniques on actual MOV equipment, while the experimental group received troubleshooting training via the MOV ATTS troubleshooting tutor. Both groups were then tested on their troubleshooting skills using a modified version of the MOV ATTS troubleshooting simulation.

Results of the effectiveness study show the trend of the experimental group (i.e., those trained with the MOV ATTS Troubleshooting module) inspected more components before attempting to correct the failure than their counterparts who were trained on the real equipment. However, a significant difference was found in the correctness of their answers. The control group made a significantly higher number of premature answer attempts than those trained with the MOV ATTS. In addition, the control group made more errors than the MOV ATTS group. The results of this evaluation suggest that the subjects trained with the MOV ATTS Troubleshooting module collected more information about a particular failure before moving ahead and correcting the problem. The control group tended to collect insufficient data before attempting to correct the perceived problem, leading to more incidence of error. Due to the small sample population in the effectiveness study, care should be taken in applying these trends to other situations.

The current phase of the ATTS project is producing an *authoring tool* to help utility instructors construct troubleshooting training systems similar to the Troubleshooting module of the MOV ATTS. This tool is the first step toward providing utilities with the means to produce ATTSs for other troubleshooting training tasks.