SCENARIO DEVELOPMENT FOR THE FIREFINDER OPERATOR AND MAINTENANCE TRAINER

RHEA J. HACKLER and ROBERT A. SEVERINO
System Engineers
Hughes Aircraft Company
Ground Systems Group
Fullerton, California

ABSTRACT

This paper analyzes the methodology used for scenario development as learned from the Firefinder Radar Trainer. First, an overall concept of the operational and maintenance trainer is given with special emphasis on the scenario training package. The training effectiveness of the system is demonstrated by reviewing the transfer of knowledge tests conducted with the actual system. The management of the scenario data base is described, stressing the use of the Chief Programmer Concept, the reuseability of the data elements, and the adherence to an effective configuration control plan. The characterization of three levels of training complexity as designed in the scenarios is also given. Next, several training schemes are suggested that make the maintenance trainer a cost-effective device. Emphasized are special CAI/CMI techniques used to transform the simulator into an effective trainer.

INTRODUCTION

The purpose of this paper is to analyze the methodolgy used to develop scenarios for the Firefinder Radar Trainer. The trainer was developed under a Naval Training Equipment Center contract, managed by PM-TRADE, for the Field Artillery School at Ft. Sill, Oklahoma. The scenario is that group of data necessary to process a student's response, cause a student station stimuli to occur, and provide error and message data for evaluation at the instructor's station. The scenario allows the student to learn in a truly advanced training environment.

Firefinder Radar Description

The Hughes Aircraft Company AN/TPQ-36 Firefinder Radar is a portable surveillance system used for detecting mortar projectiles. This radar provides first round detection and pin point calculation of battery and impact coordinates. This information is displayed on a tactical map so that a radar operator might review the data before transmitting the information to the command post via the TACFIRE link. Figure 1 shows the radar as emplaced. Figure 2 details the operational panel within the shelter unit.

Task and Skills Analysis

The tasks and skills analysis (TASA) was performed and determined the following training requirements for the radar operator and the organizational maintenance technician:

Simulated AN/TPQ-36 Operational Exercises

a. Power-on sequence

b. Computer program loading operations

c. Initialization operations

d. Hostile fire operations

e. Friendly fire operations

f. ECM countermeasures

g. TACFIRE operations

h. System shutdown sequence

i. Clearing system or on-line fault indications

Simulated AN/TPQ-36 Organizational Level Maintenance Exercises

a. Loading individual off-line diagnostic tests

b. Troubleshooting central processor unit/input output controller and memory malfunctions

c. Troubleshooting xmtr low-voltage assembly malfunctions

d. Troubleshooting SYSTEM +28 voltage distribution

e. Troubleshooting ac prime power distribution

f. Adjusting shelter, receiver/exciter, beam steering unit, and xmtr low-voltage power supplies

g. Isolating a fault to a defective low-voltage power supply

h. Aligning the signal processor analog-to-digital (A/D) converter

i. Performing the WLU map position (northing) alignments

Trainer Design Approach

Based on the TASA requirements, a trainer was developed to realistically simulate the man/machine interface found by the operator and maintenance personnel in the actual radar systems.

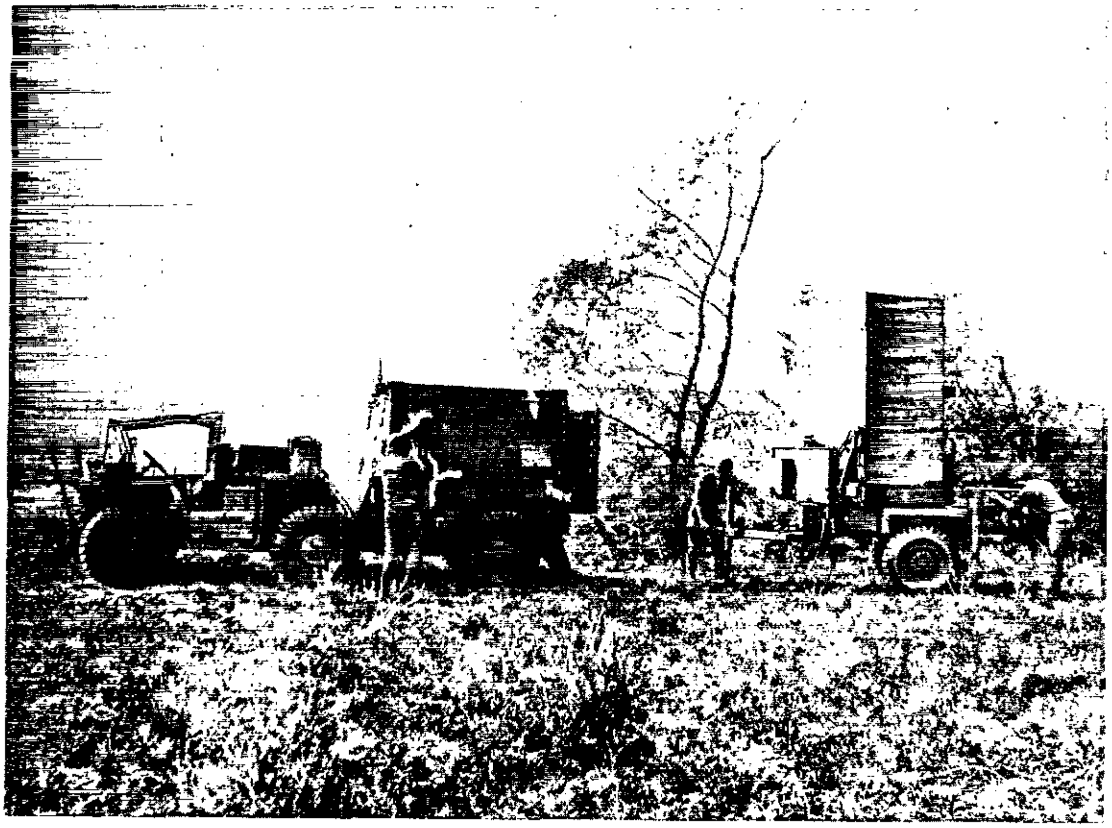

Figure 1. AN/TPQ-36 Firefinder Radar

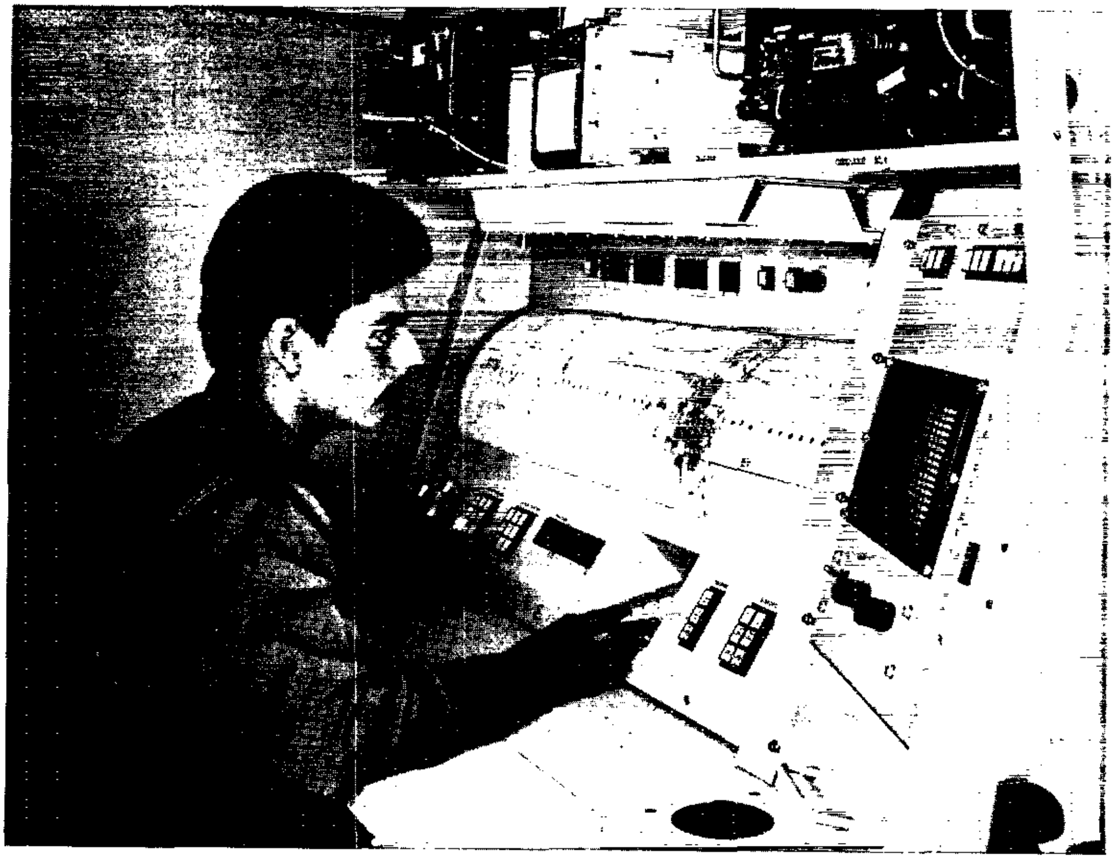

Figure 2. Radar Operational Panel

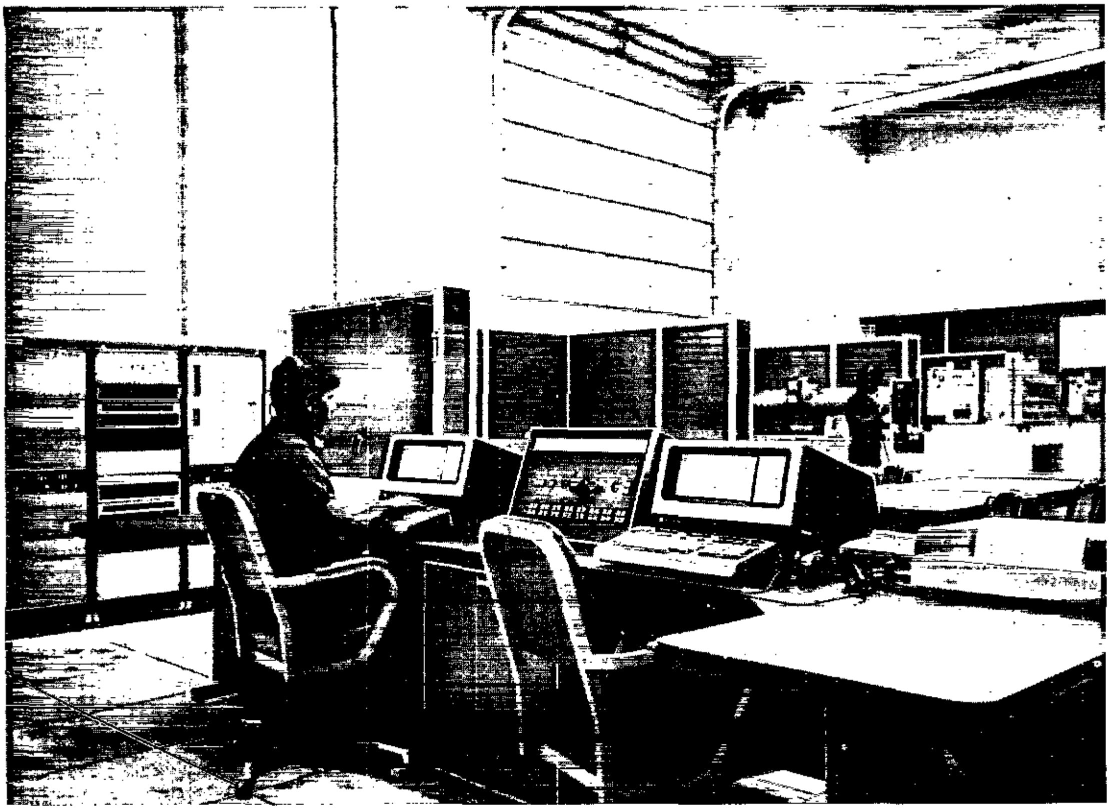

Figure 3. Firefinder Trainer

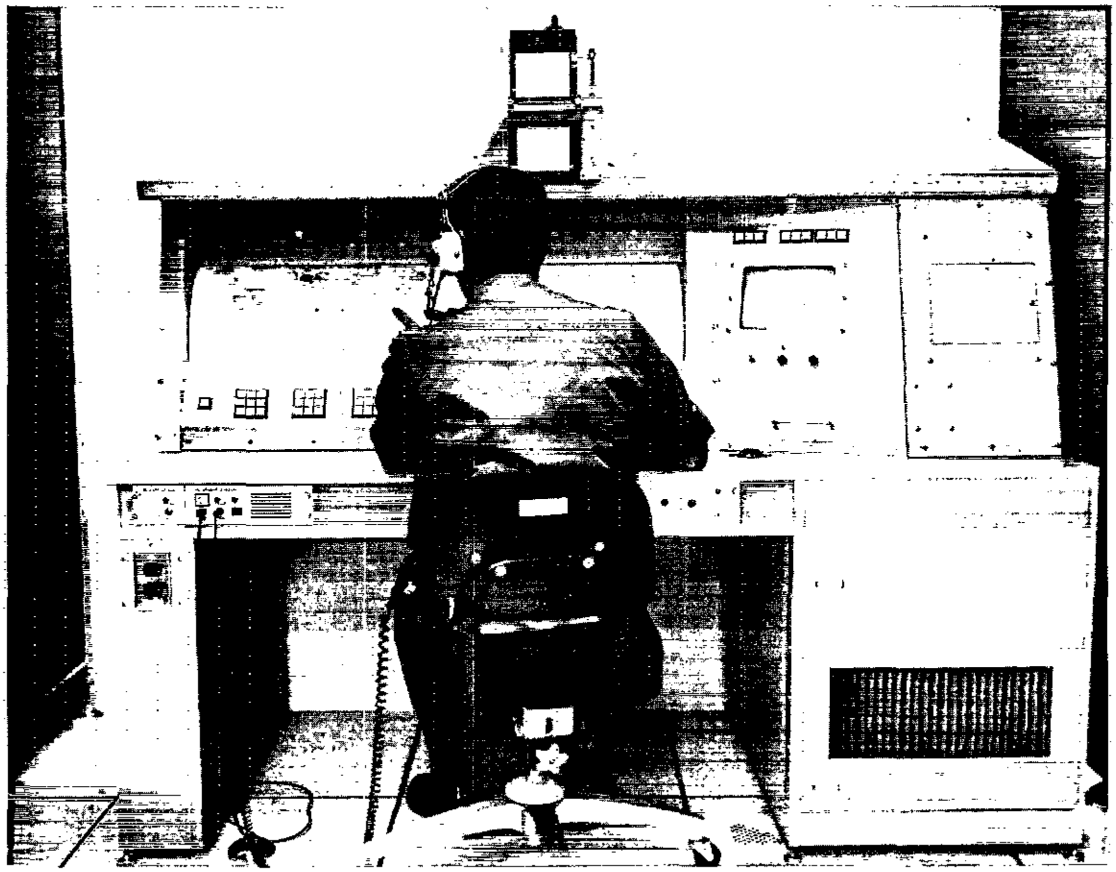

Figure 4. Operational Panel Simulator

The trainer design provides the capability for control of six student stations by a single instructor. Any combination of operator and maintenance technician students can be trained with continuous computer-managed instructor monitoring. See figures 3 and 4 for a representation of the trainer system.

The system provides the capability to simulate the Firefinder shelter and trailer indicators, switches, meters, data readouts, and other physical factors necessary to enable the desired training to occur. The student learns the various skills using a self-paced training exercise selected by the instructor to satisfy the current course curriculum. Once the exercise is initiated by the instructor, the computor will respond to each student action by simulating each Firefinder functional area.

As the students progress through the exercises, the computor provides the instructor station with student station "alert" data, and detailed "performance evaluation" data. The monitoring capabilities include detail display or printout of student actions, historic student data, and continuous display and recording of the student progress data. These features help to minimize the instructor load.

Role of Instructor

The instructor has the option of freezing the exercise, communicating any learning difficulties with the student using provided audio channels, and then continuing the exercise from the point of interruption or restarting from any point in the exercise. Any errors that the student makes are displayed at the instructor's station along with continuous updating of the student's percent accuracy, percent complete, and percent of the time standards met within the scenario-established criterion.

Software Approach

The software for the Firefinder trainer incorporates the data-base-driven approach. In this approach, the OPERATIONAL COMPUTER PROGRAM (OCP) consists of generalized processing functions, coordinated and directed by the data base. The response to any input can be altered simply by changing the data base. This data-base-driven design provides greater flexibility in defining training situations, allows diagnostics to be done on-line, and greatly reduces costs and implementation time for modification. Figure 5 illustrates the data-base concept.

Many changes may occur over the life cycle of the trainer as a result of Firefinder production changes or refinement of training objectives. Thus, update costs are a very significant design consideration. The data-base-driven design will accommodate most of the changes by data base modification which, historically, is much less expensive than software modification.

The trainer software, which makes up all the modules in the OCP, performs all of the functions in the trainer system that are not specific to a particular training objective. Examples of the functional capability of the OCP include printing evaluation reports and processing the scenario

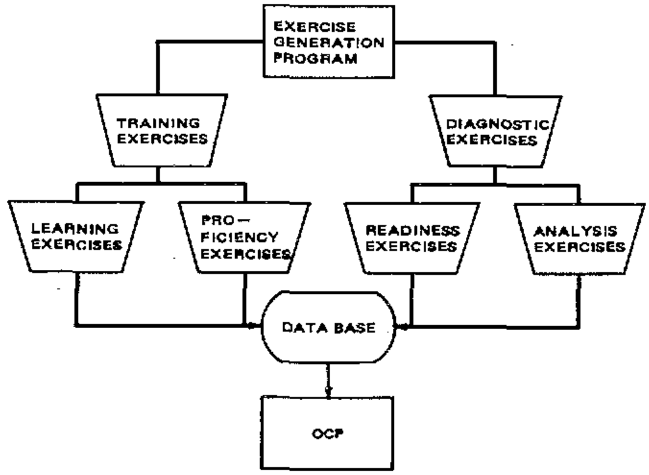

Figure 5. Data-Base-Driven Software Design

data. Figure 6 is a top-level heirarchy chart describing all of the functional areas within the OCP.

Scenarios

A scenario is a group of event forms which direct the fortran-based software package (OCP) to drive the various hardware devices. The scenario is the data that the OCP uses to process student responses, and is training objective specific. The scenario data base is that accumulation of all the training exercises which interacts with the student through the OCP. Each individual scenario will present a stimulus to the student and provide the appropriate response to his action based on the desired training objectives of the task to be taught. The scenario data base is structured in groups of families which represent the operational modes (initialization, hostile, friendly, ECM, zones and combined) and maintenance modes (equipment status tests, circuit card alignment, fault isolation tests, and power status).

FEATURES OF THE DATA BASE PACKAGE

The Firefinder trainer scenario package was developed by borrowing software development techniques already in use and incorporating novel concepts to improve the efficiency of the authorship. This section will highlight the CHIEF PROGRAMMER CONCEPT, the scenario family breakdown, the three levels of training complexity, the reuseability of data elements, and the configuration control techniques. These principles serve to increase the efficiency of the scenario creation, and aid in giving the program management office clearer visibility in the data base development phase.

Chief Programmer Concept

The Chief Programmer Concept was first developed at International Business Machines Corporation. The data base generation effort has borrowed heavily from this scheme of group organization. The scenario creation responsibilities revolve around the subject matter expert (SME), who is typically the most experienced scenario creator in the staff (In the software analogy, this would be the chief programmer).

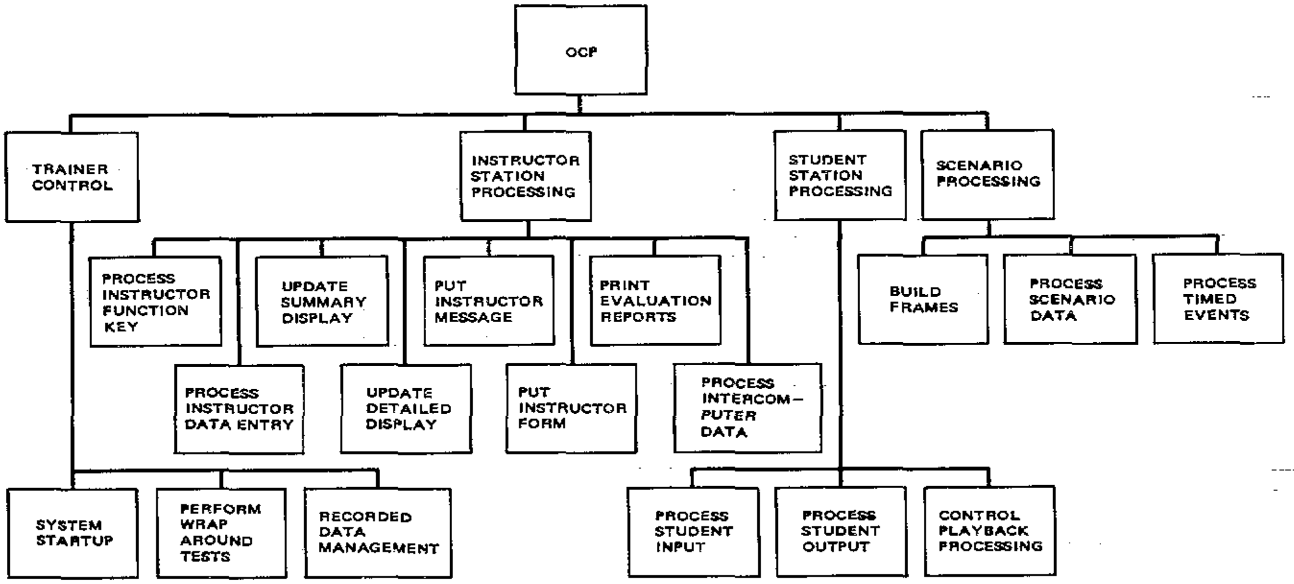

Figure 6. OCP Top Level Hierarchy Chart

The SME's job is to compile the research accomplished by others on the operational characteristics of the prime system, and develop a pivot scenario which serves as the skeleton for all other scenarios in the data base package. This master pivot design incorporates all operational emulation characteristics and excludes specific grading criteria and student timing requirements.

Other scenario creators in the group observe and aid the SME in this pivot design. After this interaction, they will have the knowledge to transform the top level pivot design into the individual scenario versions which meet the specific training objectives required of the system. Apprentice scenario creators work with and learn valuable creation skills from the SME and work directly with the other members of the group on the total data base design, instead of completing a more minor design task on their own.

Other members of the group include a librarian/technician, test coordinator, and business manager. The librarian enters all scenario data into the computer and performs other clerical and configurational assignments for the group. The test coordinator's job is to interface between the scenario creation group and the system test personnel. By observing the design methodology of the data base development, he can communicate testing criteria to the test group so that they may best validate specification adherence and exercise repeatability. The business manager has significant technical background and managerial experience. His job is to act as group manager, but not as the technical design director. The business manager typically schedules time and activities of several other groups, usually the systems test group and courseware development group. He also presents progress reports at the numerous program and division management meetings so that the SME can dedicate most of his time towards design efforts.

The Chief Programmer Concept promotes optimum efficiency in scenario authorship due to maximized reuseability of the data elements and minimum risk of duplication of effort among individual scenario creators. This results in substantial savings in core requirements and development time, thus any exercise update effort is greatly simplified due to the scenario design concepts being universally understood throughout the group.

Exercise Families

Another method used in data base generation is the family design approach. Using this scheme, all training objectives forwarded by the TASA are separated into functional groups called families. Each scenario within a family incorporates training objectives which are logically related. The SME then designs a pivot scenario for each family while the apprentice authors alter the pivot to suit the training objectives of each task to be demonstrated. The pivot is altered by adding specific data entry grading, target profiles, and customized Computer Assisted Instruction (CAI). In this way, the standard operating procedure is universal within the scenarios of a family and grading criteria can be effectively made more stringent from one scenario to the next.

The pivot for each family serves as a "free-play" scenario which does not exhibit lock-step training objectives. These scenarios allow the student to energize any switch, enter any function code, run any program, or select any feature normally found on the prime system. No grading of data entry is incorporated; however, any normally illegal action is flagged as such.

Each scenario within a family becomes progressively more difficult. At the same time each scenario is composed of 5 different options, each option requiring the student to enter slightly different data. However, each option incorporates the same level of difficulty as other options in the scenario. This provides several options to the student for reinforcement training at the same level of difficulty. Figure 7 clarifies the multiplicity of options that are available.

HOSTILE SCENARIO FAMILY

Exercises (Progressively more difficult)

Options (varied base of student entered data)	8A	9A	10A	11A	12A	.
	8B	9B	10B	11B	.	.
	8C	9C	10C	.	.	.
	8D	9D	.	.	.	.
	8E	.	.	.	.	etc.

Figure 7. Multiplicity of Exercise Options

3 Levels of Complexity

Each family incorporates 3 levels of training complexity in order to expose the student to progressively more difficult tasks. The grading categories which can exist in all three levels should be described first. The scenario can designate a student action as a not-allowed response, which is not down-graded, as it is normally a part of the actual system. This category was created to keep the student from taking alternate paths that are not included in the training objectives. Once such action is taken, the student is prompted with the letters "NAR," which forbids proceeding in the exercise until cleared from the display. Two types of error classifications exist that are down-graded. A non-critical error is that category designated by the scenario to reflect data which was incorrectly entered or an undesired action which was taken by the student. A critical error is that category reserved for switch actions which are more severe in nature.

The primary level presents few complex tasks and is characterized by a very narrow path of acceptable switch actions, with undesired paths being not-allowed. For example, the single target profile is characteristic of a primary level hostile scenario. Lenient time and grading standards are incorporated with automatic back sequencing (for example, reprompting) on incorrect student actions. Expected action messages are sent to the instructor for each step, while instructional messages referring to the appropriate technical manual page are sent to the student if an error is made. Critical errors will evoke buzzer alerts and caution messages for these potentially dangerous or harmful actions.

Intermediate scenarios are characterized by more complex learning environments, such as multiple target profiles in single mode form (hostile or friendly mode). More paths that were designated not-allowed in the beginning scenarios are now possible, and the student has the flexibility of learning new function codes that are normally a part of the actual system and is not down-graded in this exploratory training. CAI messages are utilized for incorrect responses but no automatic back sequencing is employed. CAI messages are also used in association training. As an example, the student is told at the beginning of the exercise to switch frequencies if jamming is encountered. Later in the exercise, when jamming occurs, he is graded on his ability to remember those instructions. Intermediate scenarios also incorporate tighter time standards than those used at the primary level.

Advanced level scenarios are characterized by a task complexity level equal to that which might be encountered in the battlefield. The student is presented with a combination of situations and tasks that were previously presented in singular form, that is, he is now experiencing a multi-stimulus-response arena. Typical advanced scenarios present multiple targets with jamming, priority tracks, censor zones, and interlocked hostile and friendly processing. Students are forced to make decisions based on environmental factors. As an example, he should know to change frequencies when jamming occurs, without instructional messages. All paths are allowed and total system emulation is incorporated. Time standards meet the ultimate training objectives of the course. No intervention by the instructor is available via manual back sequencing. The advanced scenarios are typically run as proficiency exercises because the instructor will be paying close attention to the percent complete field (Were all tasks performed?), the percent accuracy field (Were any errors made?), and the percent time field (Was the student fast enough?). Based on this performance data, the student will pass this family, be retested on another proficiency exercise, or be rerouted to an intermediate exercise.

Emulation

As an aid to scenario generation, the OCP incorporates an optional emulation of all switches found on the trainer. If the option is exercised, the software will automatically give a response to the student, which is the exact simulation of the response of the actual system. As an example, when a circuit breaker is turned off, the OCP will automatically extinguish all lights on the trainer which would normally be powered by that circuit breaker. This emulation option saves the scenario creator from having to define the normal response to a switch action. If the scenario creator desires a switch response different from that

exhibited in the real system, the emulation option is not used, and the desired response is specified using the appropriate equipment forms. Examples of the emulation not being used would include faulting of hardware and customized CAI.
More complex operational emulation of the characteristics of the prime system must be designed by the scenario creator. In the trainer, cues presented to the student and his responses are identical to the actual radar system and are represented uniformly to each trainee. Data for these scenarios are derived from the actual operation of the AN/TPQ-36 radar system which is currently being developed by Hughes Ground Systems Group.

Both the operational and maintenance scenarios simulate the operation of all the function codes (which are used to activate the various features of the system). The alphanumeric and graphic simulation includes B-Scope zones, header messages, function code prompts, and line priority cues. The trainer also incorporates timing precision comparable to the prime system. This can be observed in the length of the computor runs, antenna slews, buzzer characteristics, and target tracking profiles. The prime system incorporates a legality check of the operator entered data to determine if the information is in acceptable form. Illegal entries are thus flagged to the operator via the data displays. The trainer emulates these legality checks, and down-grades illegal entries as errors and then notifies the instructor. The trainer maintains integrity of all switch and lamp emulation found on the Firefinder Radar. As an example, when a switch is pressed, the scenario gives a reaction to the student in the form of lit indicators, data prompts, and buzzer cues. Included in this switch emulation is incorporated a legality check similar to that previously mentioned for data entries. The scenarios were designed so the student is able to use the prime system technical manual throughout the exercises in order to become familiar with their format and content.

CAI/CMI

While accurate emulation of the operational characteristics of the prime system is important in the design of a simulator, enhancement of the simulation is needed for training purposes. An effective trainer is much more than just a good simulator. The approach used makes extensive use of computer assisted instruction (CAI) and computer managed instruction (CMI). The CAI inherent in the scenarios gives the student simulated supervisor messages and task directions at appropriate points in the exercise. As an example, after the student has correctly initialized the radar system, he is alerted with a buzzer cue and prompted with the message "Supervisor commands you to begin radiating". It is important to understand that an operator in the battlefield cannot engage this feature without direction from his supervisor. By the scenario prompting the student, the instructor is freed from the burden of cueing the student throughout the training lesson. CAI techniques are also used in branching the scenario processing backwards to reprompt a student with the original stimuli if the student's reaction was an error. Scenario CMI is evident in the many messages sent to the instructor's console when a student has made an error. Once the instructor has been made aware of the student's difficulties, he can intervene in scenario processing by freezing, back-sequencing, zeroing (restart at beginning), or aborting the exercise.

The summary area of the instructor's display is also enhanced by an inactivity alert for each student station. If the student fails to make a switch action within a specified period of time, the instructor is alerted that the student has become indecisive and is probably lost. In the same way if the student makes a number of consecutive errors which exceed a scenario specified threshold, then the processing will send a "Consecutive Error Alert" to the instructor station. Exercise grading allows some errors to be classified as critical. These are used to down-grade students who make switch actions which might harm the equipment or cause a safety hazard, if done on the prime system. These critical errors also appear in the instructor's notice field.

The detailed display area at the instructor's station typically lists all switch actions and data entries performed at a particular student station. This display is enhanced by scenario control of two of its fields. Stimuli messages show the instructor a summary of the training conditions being presented to the student. If a student has made an error, the expected action field informs the instructor what correct response should have been taken. Figure 8 illustrates the instructor's station console and the scenario enhanced fields of information. In order to assess a student's speed in performing a training task, scenario controlled clocks are utilized to determine the time interval between switch actions. If the student's time to perform a task is longer than that required by the training objective, his time criteria score is reduced, and a notice is generated in the instructor's detailed display. See figure 9 for a CAI/CMI feature summary.

Training Effectivess Tests

The approach taken to building an effective trainer was validated by a series of government conducted training effectiveness tests (TET). The TET was performed on the operational scenarios with 20 students over 10 days under a well-structured test environment. The tests covered exercises from initialization through the various hostile modes. The first important result of the tests is that the correct number of different versions were chosen for each exercise. During the design phase, it was decided that for remedial and reinforcement purposes 5 versions of the same exercise were necessary. Results of the TET showed that a learning plateau trend was exhibited at the 4th of 5 possible versions.

A high transfer of knowledge was displayed by the students who learned on the Firefinder Trainer Device versus the control students in the test. As an example, the test group performed the initialization task on the prime system an average of 32% faster than the control group with no appreciable difference in error rates. The more interactive exercises strengthened this learning trend in an even greater way. The test group performed the hostile mode tasks an average of 47% faster than the control group and at the same error rate.

A single Firefinder Trainer device has the

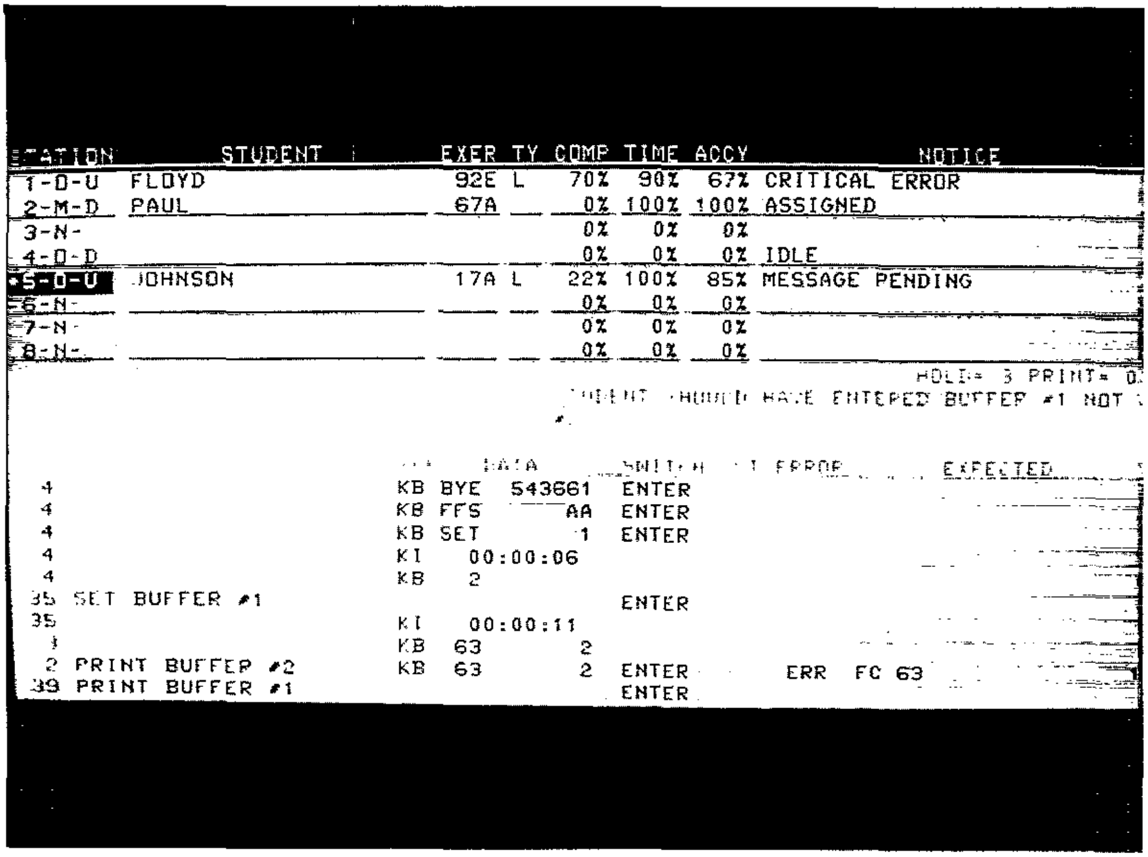

Figure 8. Instructor's Station Console Display

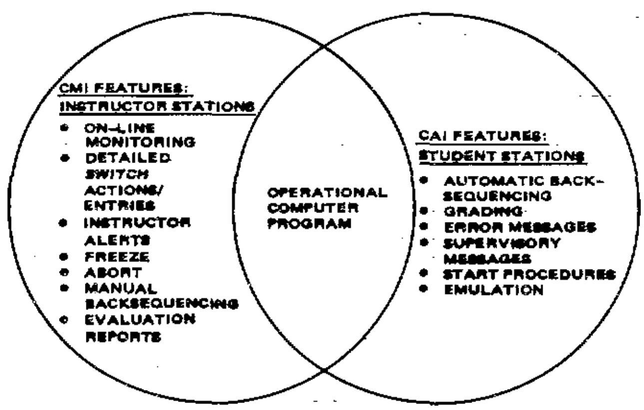

Figure 9. Dual Purpose Computer Instruction

capability of training 6 students with 2 instructors. Because of a logically arranged instructor station control format, one instructor can be allowed to monitor all 6 student stations. The TET raised the question, "Can effective training be conducted with a 1:6 ratio?" By comparing delays and performance results of 2 groups of students, it was found that students trained using the 1:6 ratio experienced as few or fewer delays and performed as well or better than the students taught at the 1:3 ratio.

Scenario Generating Program

The Scenario Generating Program is the primary support software package which provides the scenario creator with the capability to generate and modify scenario data. This program was designed to be used by inexperienced instructor personnel. Knowledge of programming languages and special skills is therefore not required. The Scenario Generating Program contains graphic representations of the locations of switches, circuit breakers, and indicators which relate to *the actual equipment. Data to be entered is in* logical terminology. For example, the scenario creator may want to illuminate or extinguish an indicator on the student station. This is accomplished by simply entering the word "ON" or "OFF" in the field next to the graphic representation of the specific indicator. Figure 10 illustrates these graphic representations.

Scenario Data Base Structure

An important factor which adds to the efficiency of the scenario data base generation is the reuseability of step elements. Before the reuseability concept is addressed, the hierarchial structure of a scenario must be covered.

One exercise contains a variable number of frames, with each frame containing a variable number of steps. Therefore, there is a three level structure inherent in scenario topography. Each of these elements (exercise, frame and step) is assigned a unique element number and is stored in a specific memory location. The scenario elements can be recalled and modified. Exercises can borrow frames or steps from another exercise, therefore, there is complete reuseability of elements at the lower two levels.

This technique reduces system storage requirement, because borrowed elements do not have to be stored twice. Savings are also realized using the pivot concept, because the scenario versions derived from the master only have to store new step elements, not the common ones. Figure 11 shows the savings realized by this reuseability design feature.

The importance of this design feature is that once a training objective is coded into a portion of a scenario, that portion can be lifted and incorporated into any other scenario that includes

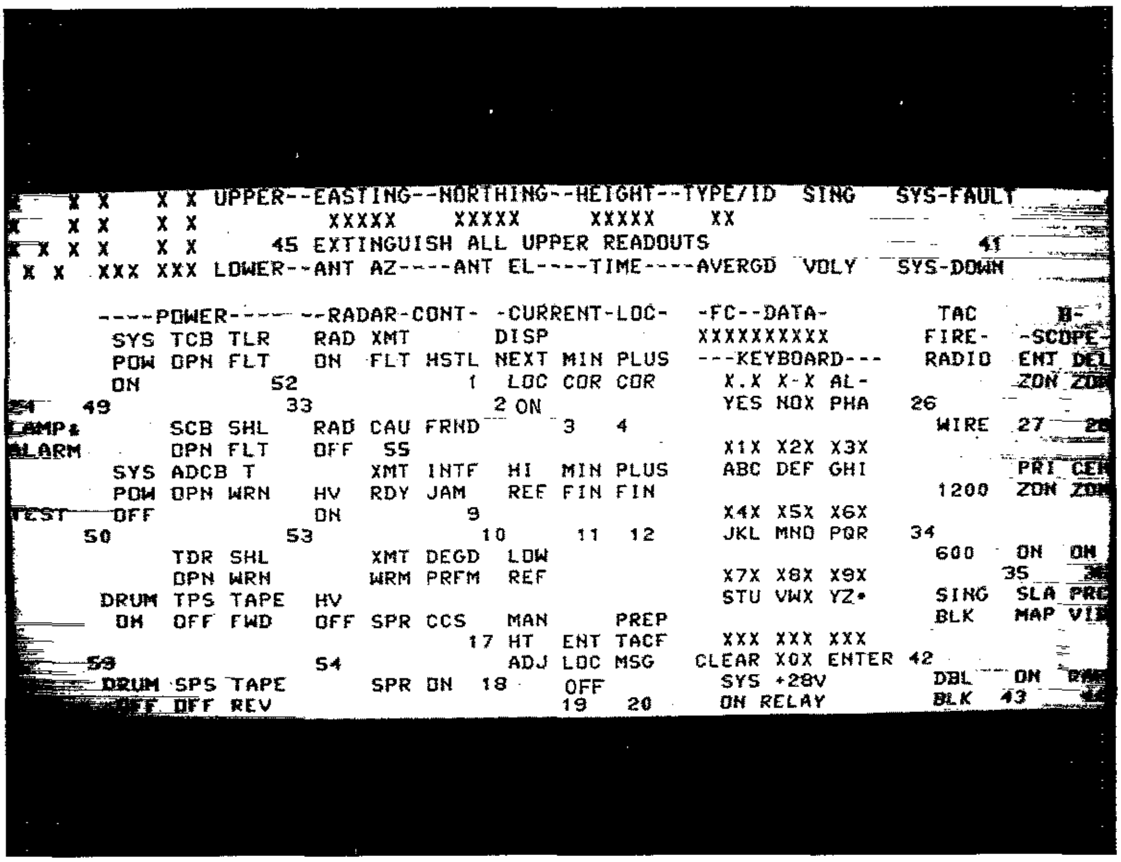

Figure 10. Scenario Generating Program Display

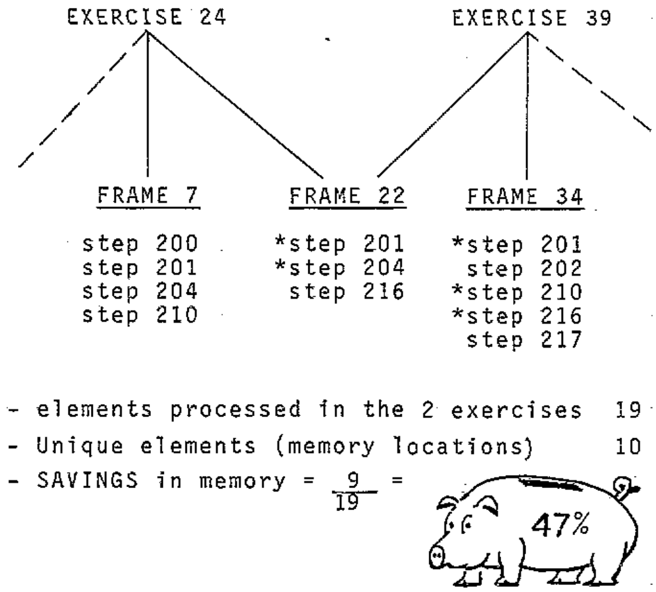

Figure 11. Savings Realized by Data Base Reuseability

the same training objective. As an example, one of the objectives of a beginning hostile scenario is to train the operator in the use of the TACFIRE prompts. A group of frames was generated to meet this objective. In the more advanced hostile scenarios, wherever the same objective was reiterated for training reinforcement, it was only necessary to copy the TACFIRE frames and integrate them into the other scenarios. In this way, the instructor personnel need not waste time reinventing the wheel as would have to be done with a more traditional software approach.

Configuration Management

The single most important tool in the management of scenario generation activities is the faithful adherence to configuration control guidelines. Because elements are assigned a unique element number, an indexing system is required to keep track of elements in use. The authoring language integral to the Firefinder trainer has this built in directory for accounting purposes. Program management requires visibility of the expansion potential of the data base. A configuration utility also exists which prints out all elements not in use.

Because elements are reuseable, many exercises might be affected by the modification of an element. A cross reference program has been developed to determine which higher level elements contain a particular lower level element. Before any modification of existing elements is performed, a cross reference is run to determine the total impact of the proposed change throughout the scenario data base.

During the generation of a scenario, the paper documentation listing is changed often. Configuration utility programs can print an entire exercise including all frames and steps in a logical sequence by sequence breakdown. This same utility can print all the steps of the data base in numerical fashion for program package delivery to the customer.

Maintenance Training

Organizational maintenance tasks taught by the trainer emphasize removing and replacing circuit cards. Card cage assemblies are represented by full sized photographic panels. Instead of removing and replacing the faulty card, the student enters a unique 3 character remove and replace (R&R) code which can be located on the photopanel above the card to be removed. (Refer to figure 12) The scenario can then grade the student's entry for correctness and eliminate the fault indications from the student's stimuli, if the R & R code entry is correct. The fault indications will remain if an incorrect card is exchanged.

In some cases the organizational maintenance technician needs to align a circuit card assembly using a potentiometer adjustment. For these cases, the trainer has provided the real pots and software emulation to simulate the normal system response to the pots being turned.

By customizing the maintenance trainer to simulate what is only needed for organizational level training, a cost-effective approach is realized. All hardware assemblies that were needed for particular training tasks were designed into the system. No extra expense was incurred by designing simulated circuit cards, which would never be needed for this level of maintenance training.

Trainer Diagnostics

To aid the instructors and trainer maintenance personnel in isolating hardware faults in the trainer, two diagnostic scenarios were provided. These scenarios perform an analysis on both the operator and maintenance portions of the student station and allow support personnel the ability to isolate faults down to the indicator, switch, or circuit card assembly. Using these diagnostic scenarios, the mean-time-to-repair of less than 30 minutes has been demonstrated.

Generic Simulation

An additional customer benefit is that the trainer has the generic ability to simulate other systems with a similar hardware configuration. By relabeling the switches on the present equipment panels, scenarios can be generated to train the student on tasks unrelated to the original design concept. As an example, a scenario to train operators for the Army's automated surveying system, "Positioning Azimuth Determining System" (PADS) was designed using the Firefinder student station. No software changes were needed, and therefore the cost for this scenario was very minimal. More elaborate training conversions are feasible by adding additional equipment panels and software support modules to the existing trainer. In this way, unrelated training environments can co-exist within the same trainer without the need for separate computers, instructor stations, or peripheral equipment.

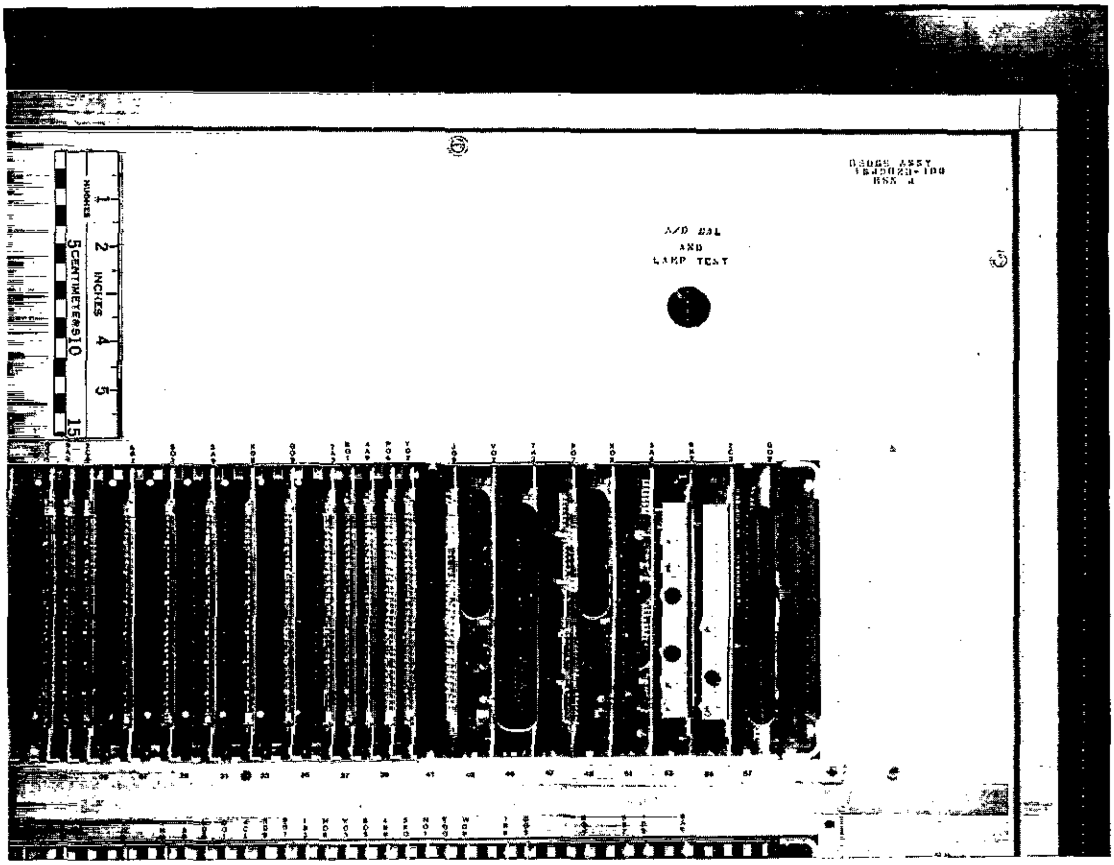

Figure 12. Remove and Replace Codes Found on Maintenance Assembly

SUMMARY

The Firefinder Trainer incorporates a scenario data package which translates each of the training requirements into an interactive system between the student and the computer. The trainer is basically a simulator, which emulates each of the functions of the prime system with a high degree of fidelity. The scenarios then enhance this simulation through the use of CAI/CMI techniques to transform the system into an effective training device. This approach was validated by the training effectiveness tests conducted. The scenario data base is easily managed because of the various configuration utility programs provided. The Chief Programmer concept and the family breakdown maximize the scenario writing staff's efficiency in developing the training programs. The further breakdown of scenarios into 3 difficulty levels optimizes the interface between the student and the instructor. The features of the maintenance trainer allow the student to increase his proficiency in repairing the system to a very high degree. The Scenario Generator Program and design of the scenario forms allow the creators to develop a training package rapidly and in a cost-effective manner. In the same way, retro-fitting the trainer to incorporate changes in the prime system is simplified. These and other design features allow the customer to realize substantial savings throughout the life cycle of the device.

ACKNOWLEDGEMENT

The authors wish to acknowledge the contributions of Major Robert White of PM-TRADE, Firefinder Training Devices Program Manager, in directing the development of the scenario data base concepts described in this paper.

ABOUT THE AUTHORS

MS. RHEA J. HACKLER is a Systems Engineer in the Training Department at Hughes Aircraft Company, where she designed scenarios for the Firefinder Trainer and worked in the development and marketing of generic trainers. As an Instructor in the U.S. Army, she created self-paced materials for the Fire Control Computer Repair course.

MR. ROBERT A. SEVERINO is a Systems Engineer in the Training Department at Hughes Aircraft Company, where he created scenarios and performed system integration for the Firefinder Trainer, and is currently working in the design of the Roland Trainer. He holds a B.S. in Engineering from Harvey Mudd College.