1-1-1989

User's Analysis And Evaluation Of Two Computer-based Training System Design Aids: OSBATS And ASTAR

James M. Ragusa

Ann E. Barron

Steven Gibbons

INSTITUTE FOR SIMULATION AND TRAINING

OSBATS AND ASTAR

User's Analysis And Evaluation of Two Computer-Based Training System Design Aids

July 1989

USER'S ANALYSIS AND EVALUATION OF TWO COMPUTER-BASED TRAINING SYSTEM DESIGN AIDS:

OSBATS AND ASTAR

Submitted by the
Institute for Simulation and Training /University of Central Florida
to PM TRADE
N61339-88-G-0002 (Task 5)
IST-CR-89-1

July 1989

USER'S ANALYSIS AND EVALUATION OF TWO COMPUTER-BASED TRAINING SYSTEM DESIGN AIDS:

OSBATS AND ASTAR

Dr. James M. Ragusa
College of Business, University of Central Florida

Ann E. Barron
College of Education, University of Central Florida

Steven Gibbons
Institute for Simulation and Training/ University of Central Florida

Submitted by
the Institute for Simulation and Training/University of Central Florida
to PM TRADE, N61339-88-G-0002 (Task 5)
IST-CR-89-1
July 1989

FOREWORD

This is the final report for Task 5, "Training and Cost Effectiveness Research in Support of Simulation and Training Technology" (contract N61339-88-G-0002 between the Army Project Manager for Training Devices and the Institute for Simulation and Training (IST) of the University of Central Florida- UCF). While the original intent of the project was to survey various cost and training effectiveness tools for training system design, the government redirected the effort to analyze two computer-based training system design tools. These are: (1) Optimization of Simulation-Based Training Systems (OSBATS), and (2) Automated Simulator Test and Assessment Routine (ASTAR).

The two major sections of the report were separately authored. Dr. James M. Ragusa of UCF's College of Business and Ann E. Barron of UCF's College of Education coauthored Section I. Steven Gibbons of IST wrote Section II. Dr. Lucy C. Morse of UCF's College of Engineering authored the final appendix. Dr. J. Peter Kincaid of IST served as Principal Investigator and Ernie Smart of IST served as Program Manager.

Special appreciation is extended to Dr. Michael Singer of the Army Research Institute, Orlando Field Unit who provided invaluable insights and assistance related to OSBATS. James Bliss of IST provided considerable assistance in developing the task lists for the VIGS and TopGun devices used for the OSBATS and ASTAR analyses described in Section II.

EXECUTIVE SUMMARY

OBJECTIVE

Training device designers require assistance in making decisions regarding how much fidelity is enough. Decision aids which standardize the design tradeoff process can control design variability and the cost of training systems while still producing effective transfer of training results. Such decision aids are not routinely applied to the training system design process although several have shown promise. The study reported in this volume provides an analysis of two training system design aids: OSBATS (Optimization of Simulation Based Training Systems), which is in prototype development, and ASTAR (Automated Simulator Test and Assessment Routine), which is ready for fielding. The objective of this project was to analyze OSBATS and ASTAR from two standpoints.

PROCEDURE

The first section describes an evaluation of OSBATS from the prospective of evaluators familiar with expert systems and government costing procedures. It is more a top level analysis than that contained in Section II, focusing on such aspects as the expert system contained in OSBATS and computer-user interface issues. The second section describes the two design aids, OSBATS and ASTAR. Evaluations and analyses are presented in some detail from the perspective of a training analyst. The application of each of the two design aids to the domain of tank gunnery is discussed. Finally, recommendations for further development of tools appropriate for training system cost and training effectiveness standardization are made. These recommendations for the OSBATS model relate to documentation, ease of use, and data requirements.

FINDINGS

OSBATS and ASTAR differ in both in their intended purpose and the way they operate. OSBATS is a much more complex decision aid than ASTAR, containing an expert system which relates instructional features to individual tasks. OSBATS also requires a large data base (differing by type of training device) including cost information. These data bases do not exist, except for rotary wing aircraft. OSBATS provides detailed guidance to the training analyst as to an optimal training device design.

ASTAR does not require a large data base and provides no cost analysis. Rather, ASTAR is a tool to assist a team of training analysts in evaluating the effectiveness of a defined training device configuration or method. ASTAR is ready for operational use while OSBATS is a prototype. Therefore, considerable attention was focused on providing recommendations for improving or redesigning OSBATS for such issues as computer-user interface, the choice of expert system, and documentation.

USE OF FINDINGS

OSBATS and ASTAR were found to serve different purposes, and in fact may be viewed as complementary rather than competing systems. ASTAR is usable in its current form, although it has no provisions for directly handling cost information. An interface between the output of the ASTAR model and a cost analysis program is needed. OSBATS was found to have considerable potential for training system design but is not ready for operational use in its current prototype form.

The reasons for continuing the development of the OSBATS prototype are valid since there can be substantial savings for training device development when the high cost of weapon systems acquisition and operator training are considered. A number of options are offered in Section I regarding the further development of OSBATS, including: (1) abandon the current OSBATS prototype, (2) redesign the present OSBATS prototype (maintaining the optimization approach) working toward a generic, more user-friendly, integrated and operational system, (3) evolve the present OSBATS prototype incorporating better expert system technology, and (4) redesign the current OSBATS using all rules and data bases that can be saved. The least expensive alternative for continuing the OSBATS effort is option (4). A framework for considering each of these options by Army decision makers is presented.

TABLE OF CONTENTS

ANALYSIS OF TWO COMPUTER-BASED TRAINING SYSTEM DESIGN AIDS:

OSBATS AND ASTAR

INTRODUCTION

The training device and simulation community has achieved the technological power to simulate training systems with impressive realism. However, this technological strength is offset when the cost and training benefits of alternate approaches and the training effectiveness of the fielded training systems are not considered. (In some cases, the cost of training devices and simulators has exceeded the cost of the operational equipment that they simulate.)

Training and cost effectiveness are critical factors in the use of simulation and training. Training and cost effectiveness variables that must be examined include measures of performance, and the degree of simulator fidelity required to produce a corresponding degree of training effectiveness (in terms of transfer to the operational equipment).

Training device designers require assistance in making trade-off decisions between fidelity and cost. Ideally, design aids for training systems should have the capability for evaluating training alternatives with respect to: (1) desired effectiveness at a minimum cost, and (2) maximum effectiveness at a given cost.

Need for Design Aids

As weapon systems become more sophisticated, the cost of training for system operation and maintenance continues to increase. The MANPRINT initiative is requiring that manpower, personnel and training requirements of alternative weapon system design concepts be accurately estimated. Early determinations of training requirements and associated resources are being made to optimize the design of the total training system; this process has a long way to go. As embedded training has taken on prominence as a training alternative, coordination between the training system and weapon system development processes has become even more important.

A number of problems in the training system development process call for a logical process of examining training trade-offs during weapon system conception. These problems include:

- Behavioral and analytic techniques to determine empirically how much simulation and training are enough need further development.
- Information must be developed on the most cost-effective use of training equipment within a course of instruction.
- Training devices and simulators have not typically been developed within the context of total training systems.
- There has been a lack of proper emphasis on the cost of training solutions, and the development of long term investment strategies for implementing these solutions.
- Generally, hard analyses of training alternatives have not been conducted because the training have not been properly measured; yet, despite the cost of training systems, no comprehensive assessment technology or performance assessment program is in place.
- The distinction between training to operate and training to fight has not been fully accommodated.

Another problem results from the many hardware options available to the training device designer. The designer's main job is to make tradeoff decisions between technology and instructional features to improve training effectiveness at the lowest cost. Many trade-off decisions are based primarily on the past experience of the designer. At times, identical design requirements can lead to vastly differing training devices. This discrepancy results from a general lack of knowledge as to which design features lead to the greatest amount of transfer of training.

Transfer of training from the training device to the actual equipment is the standard measure of the effectiveness of training device design. Much research has been conducted over the years, yet no design method has been established that ensures good transfer of training. The assumption that a training device that recreates all aspects of a system will produce good transfer results is not necessarily true; actually, a small degradation in the training device fidelity may have a small effect on transfer, yet decrease cost significantly (Hays and Singer, 1988).

Purpose of this Report

Presently, a number of tools to aid the training system design process are in development or use by the services and the DoD. This purpose of this report is to assess two design aids:

- Optimization of Simulation Based Training Systems (OSBATS), which is in prototype development

- Automated Test and Assessment Routine (ASTAR), which is ready to be fielded.

Organization

This report has two major sections and several appendices.

The first section describes an evaluation of OSBATS from the prospective of evaluators familiar with expert systems and government costing procedures. It is more a top level analysis than that contained in the second section, focusing on such aspects as the expert system contained in OSBATS and computer user interface issues.

The second section describes the two design aids, OSBATS and ASTAR. Evaluations and analyses are presented in some detail from the perspective of a training analyst. The application of each of the two design aids to the domain of tank gunnery is discussed. Finally, recommendations for further development of tools appropriate for training system cost and training effectiveness standardization are made. These recommendations for the OSBATS model relate to documentation, ease of use, and data requirements.

Appendix A is a detailed description of a user's session on OSBATS. Appendices B-F provide supporting material for Section II. Appendix G describes a way to link the output of the ASTAR model with a cost model.

SECTION ONE

AN INDEPENDENT ANALYSIS OF OSBATS

James M. Ragusa and Ann E. Barron
University of Central Florida

The purpose of this section is to provide results of an independent evaluation of the OSBATS design, useability, and supporting user documentation.

SECTION ONE

AN INDEPENDENT ANALYSIS OF OSBATS

INTRODUCTION

Overview of Section

The primary objective of this component of the task was to provide sponsors with a fairly high level independent review of Optimization of Simulation-Based Training Systems (OSBATS) software and documentation developed over a four-year period under separate Army Research Institute (ARI) contract. Since the authors were not involved in any OSBATS development, an independent and unbiased evaluation was possible. Also, the authors of this section approached this assignment from several different perspectives (education, work background, and areas of specialization).

The purpose of this section is to provide results of an independent evaluation of the OSBATS design, useability, and supporting user documentation. It is intended that the conclusions and recommendations reached will help the OSBATS sponsor and others interested in future planning, decision making, and implementation of the OSBATS concept. Without question, there is wide potential for OSBATS-type decision aid methods for training device evaluations in both private and public sector organizations. It is also realized that even small cost savings in training device analysis can result in significant overall program cost savings. A concerted effort has been made to remain independent of the analysis efforts of other OSBATS evaluators (past and present) which are documented elsewhere.

Application

This evaluation is applicable to the full range of intended OSBATS uses as described in the referenced documentation. It is limited, however, to the model's existing structure and content as described.

A BRIEF OVERVIEW OF OSBATS

Purpose

While it is assumed that the users of this study are generally or intimately familiar with the purpose and design of OSBATS, the following information is provided as a baseline for background and reference purposes. The objectives of OSBATS are described by Singer and Sticha, 1987; Singer, Mumaw, and Gilligan, 1988; and in the referenced Human Resources Research Organization (HumRRO) and Eagle Technologies documentation. These references served as the basis of the following purpose and design descriptions.

Essentially, OSBATS is a domain-specific model which is intended to provide users with an integrated decision-making process for the design of Army rotary-wing training systems. It is an attempt to organize the large body of available training technology and learning theory and to develop an implementable model for aiding training device developers in evaluating viable alternatives. As such, OSBATS serves as a decision aid that is designed to expand the number of factors that the average decision-maker considers, while increasing the speed and quality of decisions made. Version 1.1 has been developed to aid selected users in the design and evaluation of training-device alternatives that meet Army Aviation training requirements at the minimum cost or that provide the maximum training effectiveness at a given cost.

OSBATS is currently not operational and presently exists as a prototype or proof of concept system. As such it has certain depth and breadth capabilities (and limitations), and is presently designed for a specific Army system--the AH-1 Cobra helicopter. Interest in other weapon systems applications is indicated.

OSBATS, like most decision aids which have the primary objective of providing decision consistency, contains three main components: data, a user interface, and a decision system. Various AH-1 helicopter data are used, and a PC-based mouse-driven interface is provided. OSBATS is a theoretical-based decision system that uses benefit and cost approximations to analyze trade-offs between various desired training device features to identify device configurations. Once these configurations are defined, additional trade-off analysis is performed. OSBATS uses an existing Army Aviation training system, the AH-1 Airman Qualification Course (AQC), and a reduced set of training tasks as an "artificial but realistic" sample and illustrative application demonstration.

Design

OSBATS is designed as a PC-based (math co-processor required) theoretical optimization decision system. It consists of a C- language primary program supported by a run-time version of an expert system (with two knowledge bases) and a data base management program. These software programs are all required during the training device concept development process. Developed user's guide and extensive documentation are identified in the reference section of this report.

In use, OSBATS functions in two primary modes. One provides optimized results with a fixed training budget. The other uses a given set of training requirements to optimize results. The design also allows an OSBATS user to trade off the projected benefits of features relevant to specific tasks against the cost of developing and fielding that combination of features. Expert system technology (the EXSYS Corp. shell) is used to represent what is known or believed about the benefit of training device features in relation to specific tasks. A data base management system program (FOCUS by Information Builders, Inc.) is used to store all required data.

OSBATS currently facilitates the integration of trade-offs involved in developing effective training device concepts and configurations. In use, the model evaluates different training strategies (part-task training, full-mission simulator, or actual equipment training) for alternative training-device designs. These designs differ in their fidelity and instructional features and

specific allocation of training time to training devices. Five modules are sequentially used within OSBATS to make recommendations regarding the definition of task clusters, the design of training devices, and the allocation of training resources among selected training devices. These modules are labeled: Simulation Configuration, Instructional Feature Selection, Fidelity Optimization, Training Device Selection, and Resource Allocation. The modules are used in a top-down sequence until an optimized solution is reached.

Users

Singer, Mumaw, and Gilligan (1988) identify primary OSBATS users as practicing engineers from the U.S. Army's Project Manager for Training Devices (PM TRADE) organization who are responsible on a daily basis for training device concept formulation. They are described as having been involved with training device development for an average of twelve years, ranging between four and twenty years. Subsequent discussion with Dr. Michael Singer of the Army Research Institute, Orlando Field Unit, indicates that in addition to these 30 or so engineers, two other groups are candidate users. The first are 15-20 PM TRADE logistics engineers who are responsible for the ultimate implementation of training device decisions. The other potential users consist of resident training developers at U.S. Army Aviation Centers such as Ft. Rucker, Alabama.

OSBATS EVALUATION

Systems Analysis and Design

Background The use of systems analysis and design concepts and techniques is a standard part of data-processing, management information system (MIS), and decision support systems (DSS) development. For that reason these techniques are widely taught in colleges and used in practice. There are numerous descriptions of the process, and fairly standard elements (with variations) are usually identified. Elements of what is called the systems development life cycle (SDLC) have been identified by Kendal and Kendal (pp. 8-13). These elements are:

- Identifying problems, opportunities, and objectives
- Determining information requirements
- Analyzing system needs
- Designing the recommended system
- Developing and documenting software
- Testing and maintaining the system
- Implementing and evaluating the system.

Two approaches to the systems design and analysis process are identified by Ostle (p. 187) as consisting of a top-down structured design and prototyping. Top-down structured design works well for small and large computer systems only if: (1) the application is not particularly unusual, (2) the developer has worked on similar systems, and (3) no surprises crop up. Conversely, prototyping: (1) is quick-and-dirty, (2) involves trial-and-error, (3) is often effective, and (4) is a way to get systems operating with a minimum of delay. Use of this latter approach offers expectation that the results will serve as a model that will need further, and perhaps extensive, modification and development.

Importantly in both cases, users need to be involved in the SDLC if the system is to evolve as a viable, useful, and accepted system. Numerous examples exist of situations where users were not part of the development, and it subsequently failed due to inadequate performance and/or lack of user acceptance. The key to SDLC success, as reported in the literature, is that users need to be intimately involved in all phases of development activity from beginning to end. In practice, those who use the prototyping approach seem to recognize this essential participation better than those who follow the more traditional top-down approach.

Unfortunately, there is a lack of strong evidence in OSBATS documentation and related articles that PM TRADE users actively or continuously participated in the development process. Without that participation it can be expected that reluctance to use OSBATS might result. Additional discussions concerning user acceptance are addressed in other areas of this report.

An Important Expert Systems Variation Those involved in expert systems construction have gone one step further than conventional system builders by using what is called rapid prototyping. Here domain (working environment) experts and users (who may not be the same people) participate at every stage of system development. Edward Feigenbaum, a prominent expert system researcher, suggests that a philosophy of a "peek in a week" be used by rapid prototype developers.

The use of the expert systems rapid prototyping methodology can be compared to the traditional system analysis and design software development cycle (and in software engineering in general) in the following way. In the conventional SDLC environment, the way software is developed is through the identification of a requirement specification by a systems analyst and the user. The systems analyst then meets with the system designer and develops a functional specification. The system designer then develops a design specification and gives it to a programmer, who then implements it. With large software projects, there may be several months or years between the time when the requirement specification was developed with the user's aid and when the user first

sees the result of his discussions. The typical reaction of the user upon first seeing his system is, "You totally misunderstood my problem. This is not what I wanted. I don't like this. Go back and change it." So you then take your beautiful design and your beautiful system, and you start putting "warts" on it until the user is willing to accept it.

In an expert systems environment, a totally different approach is taken. The job functions of the systems analyst, the system designer, and the programmer are combined, and a new function called the "knowledge engineer" (KE) is created. The KE meets with the user, and together they develop a requirement specification after considerable discussion. But then the KE starts experimenting, creating a small prototype or demo of the system. The user then sees this demonstration and immediately says, "That is not what I wanted. You totally misunderstood my requirements. Go back and change it, or throw it away, or rewrite it." Exactly the same reaction that occurred during the development of the conventional system. The difference is that there were only several days or weeks before he saw something in the ES rapid- prototyping approach. The KE then goes back and changes the system and continues adding more functionality and correcting errors, gradually developing a full expert system.

The level of improved communication with the KE, the expert, and the users prevents having to take twenty steps back. Instead you may only have to take one or two steps back. When the system is fully implemented, the user has participated in every phase of the design and every phase of the development. The system was designed as it was developed, cutting out the design step, with considerable time saved as a result. Moreover, the first system delivered is correct, which greatly reduces the final delivery time of the system. Also, a correct system is much easier to maintain than a system with warts on it. The total result is that a savings of usually five times in productivity is achieved through the elimination of the design cycle, the reduction of maintenance costs, and the more effective development of the correct product.

How is expert systems rapid prototyping different from rapid prototyping using fourth generation tools? The principal difference is that with expert systems the capability exists to do object-oriented (frame based) programming. While discussion of this type of expert systems programming is beyond the scope of this study, it is important to note that when used in a knowledge engineering environment, it is possible to develop code that has considerably less software side effects and interdependencies between different software modules. Greater flexibility also exists to manipulate entities without worrying about how the entities are going to be used later on.

User Interface

While the intent of this report is to address higher-level issues, the topic of useability is important. The following user specific interface comments were developed during the OSBATS evaluation process. "The user interface is a critical part of any decision aid, and serves as the basis for user understanding and confidence in system processes and recommendations" (Singer & Sticha, 1989, p. 28). For the purposes of this report, the OSBATS user interface was examined using several guidelines set forth in Ben Shneiderman's book, Designing the User Interface: Strategies for Effective Human-Computer Interaction. These examinations include the areas of input devices, functional areas of the screen, data display, screen colors, menu structure, help screens, response time and display rate, and error messages.

Inputs Devices OSBATS is primarily dependent on mouse inputs. This approach reduces the chance of typographic errors on the keyboard and allows the users to focus their attention on the display. The mouse inputs are "appealing because the hand rests in a comfortable position, buttons on the mouse are easily pressed, even long motions can be rapid, and positioning can be very precise" (Schneiderman, 1987, p. 242). "The results are often faster performance, fewer errors, easier learning, and higher satisfaction" (Foley, VanDam, 1984).

Data Display Smith and Mosier (1984) offer five high-level objectives for design of data displays. They are:

- Consistency of data display
- Efficient information assimilation by the user
- Minimal memory load on user
- Compatibility of data display with data entry
- Flexibility for user control of data display.

Each objective is discussed in relation to OSBATS as follows:

Consistency of Data Display -- This guideline deals with standardized terminology, formats, and abbreviations.

The major inconsistency found in OSBATS is the titles (labels in the headers). Good design techniques demand that all screens be titled. In most cases, in order to provide a clear path for the user, menu selections become the title of the screen to which they branch. In OSBATS, not only are some screens untitled, but the menu access to the screens is very inconsistent.

As an example, if you choose DISPLAY RESULTS from the Simulation Configuration Menu, you branch to a screen with no title. The header says only: CURRENT TASK SET: If the user branches to a lower screen, such as TABULAR SUMMARY, GRAPHICAL SUMMARY must be chosen to get back to the DISPLAY RESULTS screen. This is extremely confusing for the user and occurs on a majority of the screens.

A similar problem exists with the terminology used for the menus. The root (main menu) is entitled MODULE SELECTION MENU. From lower menus, one may branch back to this menu by choosing MODULE MENU. The next level of menus (Simulation Configuration, Instructional Features, etc.) are referred to by sub-menus as either the MAIN MENU or the MODEL MENU. In other words, choosing MAIN MENU from EXAMINE/EDIT TASK SET will branch to the INSTRUCTIONAL FEATURES MENU. However, from the DISPLAY RESULTS/RESULTS TABLE menu, one must choose MODEL MENU to branch to the same INSTRUCTIONAL FEATURES menu. These issues (no titles and inconsistent terminology) are extremely confusing for the user and could be corrected with minimal effort.

Efficient Information Assimilation by the User -- This objective is served by rules for neat columns of data, left justification of alphanumeric data, right justification of integers, lining up decimal points, proper spacing, comprehensible labels, and appropriate use of coded values.

These techniques are incorporated very well. The only suggestion would be on-line help for some of the terminology used. In other words, if the user is unfamiliar with the term SCAS Off Flight, there could be an on-line dictionary of terms that could be accessed.

Minimal Memory Load on User -- This involves techniques that do not require users to remember information from one screen for use on another screen. Tasks should also be completed within a few commands, minimizing the chance of forgetting to perform a step.

A major drawback of the OSBATS system is that it relies on the user to remember and compare several different approaches. Even following the user's manual, it is hard to comprehend the significance of different approaches. Perhaps some sort of comparison mechanism could be built into the system that would keep the "conclusions" and provide a comparison for the users.

Disadvantages of using a mouse for the input device include the requirement to take your hands off of the keyboard and the increased cognitive processing and hand-eye coordination required to locate the desired target. The mouse also consumes desk space, "the mouse wire can be distracting, pickup and replace actions are necessary for long motions, and some practice is required to develop skill" (Schneiderman, 1987, p. 242).

In general the use of the mouse for input in this application is appropriate, especially if the user is comfortable using a mouse. The targets are much too small for touch screen input and too far apart for cursor selection. However, the following problems inconsistencies are noted in the program:

Cursor -- The cursor is generally a small arrow that is magenta in color. This color changes to a dark blue when it enters an active field (designated by black foreground and yellow background). Because the dark blue is so close in color to the black foreground, it is often difficult to determine the cursor's position on the screen. This confusion could be reduced by either using a different color or a larger shape for the cursor.

Keyboard Entry -- There are some inputs that demand keyboard inputs. This requires the user to take his/her hand off of the mouse and position it on the keyboard. If a file name, etc., must be entered, this switch to a keyboard entry is necessary evil. However, there are instances where the input required is a simple "Yes/No" (y/n). This could easily be achieved by using the mouse. (See the Problem Solution Selection Menu in the Resource Allocation Module.) In fact, a mouse input would be better because a "y" or "Y" is judged as a "Yes" and all other inputs are judged as "No".

Return -- A similar confusion results when the single input of "Return" is allowed on a screen (such as the Help screens). It is a natural reaction to press the "Return" key in this case; however, the required input is a click of the mouse on the word "Return". To eliminate this confusion, the system could be altered to accept either the mouse or keyboard input, or the message could be changed to "Click here to return".

Screens -- Some of the screens have a series of items that can be clicked on to transfer them from one side to the other. This method of control works very well. However, on some of the screens, multiple items can be moved by sliding the mouse down the screen. On other screens with an identical format, each item must be clicked on individually. This inconsistency can cause great confusion.

Functional Areas of the Screen The use of functional screen areas is quite consistent. In general, the top four lines contain model status information and other pertinent facts such as the current task cluster and the status of the weight values. The middle 31 lines present the graphs or tables of facts, and the bottom seven lines contain the available menu options.

The use of the menu options at the bottom are quite clear and consistent. If the option has a yellow background, and a black foreground, then it is active. If it has a black background and yellow foreground, then it is inactive or currently in use. There are no on-screen instructions regarding interaction with these options; however, it is covered in the user's manual. The functional area at the top of the screen is a bit confusing. On some screens there are active areas at the top, and on others there are not. In addition, the impact of "clicking" on the options in this area are not as clear, leaving the user confused as to what exactly happened. Again, there are no on-screen directions. The area in the center of the screen may or may not have active areas. Some, such as the Graphical Summary screen in the Simulation Configuration module have asterisks that can be clicked on to obtain information. Other screens containing asterisks are not active. This is explained in the user's manual; however, on-screen instructions would be helpful.

Compatibility of Data Display with Data Entry -- The format of displayed information should be clearly linked to the format of the data entry.

The OSBATS system attempts to deal with a wide range of users through the use of graphs and tables to present results of the trade-off analyses performed and the information used in making the analyses. This provides the user with different ways of viewing the results.

One limitation on the flexibility of data display in OSBATS is noted with the viewing of the tasks. On most of the task screens the user has the option of NEXT TASK or PREVIOUS TASK. This requires the user to navigate through all the tasks to arrive at the desired one. Perhaps an option could be added that would allow a direct branch to the desired task.

Flexibility for User Control of Data Display -- One item that seems to be lacking in the system is the ability for experienced users to take shortcuts to a particular location. Special keys, hidden commands and/or macros facilities are all examples of possible methods for incorporating shortcuts for frequent knowledgeable users. OSBATS does not include any of these techniques, requiring the user to traverse through all of the menus.

Screen Colors In most cases, the colors used on the screens are consistent and well-chosen. The basic colors are dark blue, light blue, yellow, and black. Except on the bar graphs, the only other colors used are green and red. Most screens use less than the generally accepted limit of five colors per screen. An exception to this is the bar graphs, which of necessity use as many colors as are available. Other than the problem of the blue cursor disappearing on the black foreground, the colors are quite appropriate. For instance, the red color is used only to indicate a selection marked for deletion, and yellow indicates an active area.

There are some "non-system" screens that are strictly white text on black back-ground. These are used for selections such as "Save Results" and "Select Problem." It is unclear why these screens differ from the other menu selections.

Menu Structure The menus are organized in a tree structure, based on the five modules. Each module then has three or four layers of depth. There is no clear path or structure for the user to use to determine his/her position within the framework. This could be alleviated by providing a flowchart in the user's manual, an on-line graphical design structure, or a numbering system in the menus. The problem with the inconsistencies of the screen title (see section on Data Display) further complicates this issue.

To allow for ease in navigation through the system, all menus should provide access to previous menus and the main menu. (Shneiderman, 1989, p. 111). In OSBATS, all of the menus permit jumps to the previous menu. Some of the menus permit jumps to parallel menus. Most of the menus do not permit direct jumps to the main menu, requiring the user to retrace the steps up the tree to the main menu.

The sequence of the menus is also important. In this case the menus are sequenced based upon the five main modules. This is an appropriate sequence, but it is unclear to the user which, if any, of the menus can be skipped.

Help Screens Help screens are available from most of the system screens. The help screens are less than one page long. This is good, in that it prevents the need to scroll. However, it does place excessive text on each page, and the resolution often causes the lines to almost overlap each other, making reading difficult. To solve this problem, it would be useful to highlight some of the keywords. Currently, all of the text in single-spaced, with a white foreground and black background.

Because none of the screens contain instructional information, the help screens are especially important. They would be even more "helpful" if the explanations of menu options were highlighted in a different color and placed in a discrete location in order to provide quick viewing.

The help screens also seem to have an excessive amount of grammatical and spelling errors.

Response Time and Display Rate Most of the screens have an appropriate response time and display rate. The major exception to this is the Display Results screen in the Simulation Configuration Module. This complex screen takes considerable time to display and can be quite distracting. One solution might be to load this screen into memory so that it does not have to replot each time the user returns to it.

Error Messages The system contains very few error messages because it is primarily a selection/comparison system. Unless one "clicks" in an active area, the system does not respond. One situation where the user is instructed to enter a "User Defined Name," the error "Too many chars" is generated if more than eight characters are input.

User Acceptance

When a new computer system is developed and implemented, hardware and software test plans are regularly included in requirements documents; extending that principle to the human interface development through an acceptance test is a natural and important extension. Five basic features are considered by Shneiderman (1987) as essential to user acceptance. These features are: time to learn, speed of performance, rate of errors by users, subjective satisfaction, and retention over time.

Time to Learn In order for users to feel comfortable with a new system, they must be allotted the time and materials with which to learn it. In a system as complex as OSBATS, this training time could consist of several days of instruction. The instruction could be in the form of lectures, computer-based tutorials and simulations, or actual "hands-on" training with OSBATS. A series of practice scenarios with feedback from an instructor or computer should also be included to allow the new user to gain confidence in the application.

Speed of Performance If a new system can reduce the amount of time required to perform an operation (such as selecting an appropriate training system design), the users will be more inclined to implement the system. Studies should be undertaken to analyze the speed with which OSBATS can configure a training system as opposed to the traditional approach.

Rate of Errors by Users Another way for a system to "sell itself" is for it to increase productivity and/or decrease errors. In order to prove that this is the case with OSBATS, studies should be undertaken to evaluate the accuracy of the recommendations made by the system.

Subjective Satisfaction One way to promote a new system is to provide research that points to its effectiveness and advantages. Another method is to allow the users to convince themselves by providing a system that is user-friendly and satisfies their needs. For instance, if the user tries a new system, and receives appropriate conclusions, the user will be more likely to adopt the system and use it in the future.

Retention Over Time In addition to the initial training time allotted for learning OSBATS, it is important that the training and procedures be retained and not require re-learning every time the system is used. The training methods and media can have an impact on retention. For example, interactive video training has been shown to have a much higher retention rate than other types of instruction, such as lecture, video, and computer assisted instruction.

User acceptance is critically important for the implementation and maximum use of a new computer system. In addition to the five factors previously mentioned to enhance user acceptance, the user should have access to avenues to obtain assistance with the system. These avenues could be in the form of a user "hot-line," surveys (followed by quick response time), or a users' group with expert input available.

Algorithms

Background Since World War II operations research (OR), with its toolbox of techniques, has evolved from various real-world needs and mathematical discipline contributions. As a result, OR has been a popular educational and research field of study. While not as widely adopted as hoped, numerous OR applications have been found in various business, scientific, military, and aerospace domains.

One advantage of the use of OR as a decision aid has been the exactness of resultant optimized decision aid solutions. Another is the building block framework which has allowed the inclusion of new techniques as they develop. The primary problem with the OR methodology has been that frequently assumptions (sometimes heroic) must be made about the environment under analysis to make the problem solvable. Another problem has been user acceptance, associated with the fact that most users do not understand the technique(s) used to find the solution. Both of these problems have resulted in a lack of full acceptance of OR as a solution methodology. Too often the result has been lack of full use or avoidance of either techniques or results (even after management decree). Without this acceptance, regardless of reason, even the most exacting, elegant, and optimized solutions will be ignored.

Evaluation The optimization methodology selected for OSBATS requires extensive use of various OR algorithms. Learning curve, Pareto optimality, clustering techniques, and sensitivity analysis are a few of the methods employed. Of significance is the selection and use of the Integrated Computer-Aided Manufacturing Definition (IDEF0) system analysis method. IDEF0 provides a top-down analysis of model components and their relationships.

Without question the algorithms and system analysis methods used in OSBATS are very sophisticated, complex, and well done. However, to a great extent they represent a black rather than a glass box--especially from a user's perspective. Except for the designer(s), it is almost impossible for anyone to understand what is going on in that box. The resultant solutions may be optimized, exact, correct, and elegant. The question is: Will the user(s) believe them? Even a top-level IDEF0 flowchart would not provide a map for building confidence that the numerous paths through this extremely complex network are correct or consistent with the existing method used by PM TRADE engineers or training center users to reach training device solutions. Thus a dilemma exists: Will users believe and accept OSBATS or continue using their existing individual and customary method of analysis? Historically, unless a very strong reason or motivation or understanding and trust exists, users will default to what they know and are comfortable with. It could be logically concluded that OSBATS in its present form will not experience confidence and acceptance for the above mentioned reasons.

Expert Systems

Background Expert system technology has captured the imagination of the press and general public over the last ten years. Still, few understand the technology and the system building process, and even fewer understand what applications are possible. Regardless, more and more systems are being built to computerize repetitive, complicated, and time-consuming decision-making chores.

An expert system can be viewed as a computer program capable of automating decisions usually by the use of question responses by users. Through this process, answers, conclusions, recommendations, or tabular data are provided. In reality an expert system shell is a software program which has all of the structural elements needed to build a system like a pre-made pie shell prepared to accept ingredients. In the business world, spreadsheet programs are shells which have the necessary structure to allow a user to quickly build customized "what-if" decision aids. Expert systems also have one very significant advantage over more conventional software in that they have explanation capability. This inherent feature, which is very important to decision makers, is that part of the system that explains how conclusions were derived and the steps used to reach them.

Expert systems are a departure from the more classical quantitative techniques associated with OR and the algorithms used in solution methods. The difference is primarily that of optimization vs. satisficing. While both are considered quantitative techniques, they have significant differences. An optimizing technique provides the optimum or best solution (e.g., maximum or minimum value) while a satisficing technique provides a satisfactory but generally suboptimal solution. According to Herbert Simon (1957) economic man maximizes -- selects the best alternative from among all those available; his cousin, administrative man, satisfices -- looks for a course of action that is satisfactory or "good enough" or is "better than before." Expert systems rarely optimize but instead find solutions which satisfy the constraints and conditions of the problem. This difference can have significant differences in cost and approaches to problem solutions.

Building an expert system usually requires at least three people: (1) a domain (environment) expert whose expertise (knowledge and heuristics) will become the "heart and soul" of the system; (2) a knowledge engineer (KE), who will structure the captured knowledge in a more usable form; and (3) the rule builder who will actually construct the expert knowledge-based program. Because of the ease of use of commercially available shells, one person may perform all or multiple roles during system development and implementation. However, shells and systems are still being built using the traditional role structure, particularly if the creators desire maximum control or if the system is large, requiring the special knowledge of a non-computer-oriented expert.

While several expert system shells are now available, the first commercial expert system shell developed was E-MYCIN. This shell evolved from a much larger medical blood disease diagnostic system built at Stanford University in the early 1970s. E-MYCIN was the "Empty" MYCIN program which remained after the domain- specific blood disease knowledge base was removed. Since that time, several commercial shells have used the E-MYCIN structure as their basis.

Already mentioned is the KE--both a person and a process. The KE is the bridge between the domain expert and the expert system hardware and software. Involved in these activities are: knowledge acquisition, system analysis, knowledge structuring and representation, software and hardware selection, application design, and coding. If done properly, these activities will result in a significant improvement in the understanding of the total problem under development.

Use of EXSYS EXSYS is one of approximately ten popular rule-based shell programs. Written in the C-language for high speed and efficient use of memory, it is a full capacity system which allows interface to spreadsheets (Lotus 1-2-3) and data bases (dBASE III+ and RBase). Over 5000 rule systems can be constructed using EXSYS, and forward and backward chaining rule processing is possible. The shell normally requires only 640K of RAM.

EXSYS is used in OSBATS to develop and process two rule bases: one for instructional features and the other for fidelity requirements. The former rule base consists of 33 rules while the latter in considerably more extensive and contains 237 rules -- many of which are compound in nature. The size of these rule bases is small compared to almost all operational systems but is representative of prototype systems.

According to the OSBATS User's Guide, the instructional feature selection module uses a set of rules, which has been developed through research, to compute the benefit of a task-by-task basis for each available feature. The feature benefits are compared to their respective costs, and a benefit-to-cost priority list of features is developed. As a result of consultation, a file consisting of a two-dimensional matrix of zeros and ones is created. These values indicate which instructional features are appropriate for training each task. The far more complicated fidelity rule base is used to develop cue and response requirements. It contains rules about use and response requirements for training that task. The generated file is a table of cue and response requirements organized by task and fidelity dimensions.

Evaluation The use of an expert system shell as part of OSBATS is valid -- in fact, as will be discussed later, OSBATS did not go far enough in taking advantages of the power and functionality of expert system technology. The tasks both rule bases perform are complex, and unlike other manual and automated systems both programs can result in quality results. Extensive work has gone into their development -- especially the 237 rule Fidelity rule base.

Several comments are made concerning improvements which could be made in the future. They are as follows:

Integrated System -- The most serious concern is that the users does not have ready access to either rule base in OSBATS. The user must leave the CODE subdirectory to exercise each rule base. These rule bases as such are truly not part of an integrated OSBATS program. Therefore, users do not have ready access to the explanation capability inherent to expert systems (including EXSYS). The obvious advantage of an explanation capability is that it does much to provide logic, an audit trail, and creditability for the user's decision process.

Knowledge Engineering -- It is not very clear what research or knowledge was used to develop the rules used in the two rule bases. As mentioned earlier, usually (but not always) domain expert knowledge and heuristics are the basis of rule base developments. In the case of OSBATS development, a choice was made not to use one or more of the PM TRADE user "experts."

EXSYS -- While EXSYS has been fairly successful in the past because of its features, the OSBATS designers did not use the capabilities it has to integrate with other programs such as data base programs. These hooks are of great value to building integrated and user-friendly programs. Also, a better shell choice now would be LEVEL 5, a program presently made by the same company (Information Builders Inc.) that sells FOCUS (the OSBATS data base program). Both programs are fully compatible and can be integrated easily. While colors are essential to the use and marketing of expert systems, a better color selection is needed. Many colors are too dark, would be unreadable for colorblind users, and could not be displayed on recently commercialized computer/viewgraph projection equipment if used for training. Help screens were non-existent or lacking.

Data Base Structure

Background The data base use for OSBATS is called FOCUS from Information Builders Inc. This system is a fourth generation language and data base management system (4GL/DBMS). According to the vendor, FOCUS is a powerful language for developing applications which offers automatic generators for producing application code, screen layouts and presentation systems with windows. This software system allows users to have the tools necessary to build computer applications for reporting and managing data -- without requiring other tools or third generation languages. Importantly, FOCUS is very portable and runs on a full range of computers from PC-based systems to mainframes and large minicomputers. Data can be uploaded and

downloaded to a mainframe host if needed. In addition, FOCUS allows users to use inherent 4GL tools with a wide variety of other sources of data (e.g. dBASE, RBase, Oracle, DBMS, etc.). Other application development facilities include:

- A rich native language.
- Pop-up or pull-down windowing.
- Intelligent defaults.
- Procedural controls.
- Screen printing.
- Interactive debugging tools.
- Host language interface.

Evaluation The selection of FOCUS as the data base system by the OSBATS designer was good -- with reservation. FOCUS is an established system which is widely used and well supported by the vendor. Applications developed by users are widely varied in scope and dimension.

There are three primary problems in the relationship between FOCUS and OSBATS. The first is that, as best as the authors can determine, the data base is not complete for the AH-1 system. Secondly, a user cannot access the data base within OSBATS for additions or deletions -- if there is a valid reason to do so. Finally, FOCUS is not compatible with the EXSYS expert system shell and developed knowledge bases.

In summary, what exists is a non-integrated critical subsystem of an optimization model which is not complete, which the user cannot change, and which does not directly interface with the expert system rule bases that contain the facts and heuristics for the domain. This would lead to the obvious conclusion that systems design and user interface shortcomings exist with the present prototype.

Other Considerations

Throughout the study observations and considerations were reached which did not seem to fit into categories of the report. In lieu of just disregarding these thoughts the following are listed:

Cost Data -- There was little evidence that present value or cost inflation factors were used for future periods. One could assume that costs used were inflated. To be realistic, optimization model should provide a means of taking into consideration the time value of money.

Life Cycle Costing -- The use of life cycle costing was evident. Perhaps this was required of the developer by federal rules, however, it is a very enlightened and cost sensitive approach. Too often, front-end cost loading or ignoring life cycles lead to unrealistic cost conclusions.

Parallel Processing -- If future OSBATS-like programs become too large or generic, processing on a workstation or mainframe may not be the answer. Perhaps use of a parallel processing machine (with its inherent speed improvements) would be more appropriate. OSBATS-type applications seem to be compatible with parallel processing concepts where portions of complicated algorithms are distributed to separate processors. Networking in this manner appears consistent with the form and function of the IDEF0 methodology which describes the functions and data of a complex system. Unfortunately present parallel processing machine technology, while existing, is expensive. Fortunately, as has been the case with computer technology, in time costs and capabilities will make this kind of a platform cost efficient.

Interactive Video -- There appear to be several instances where visual images or motion sequences could be tied into OSBATS to provide visual assistance to users. There are numerous instances where interfaces between mass storage systems (text, visual, and audio) and procedural or intelligent front-end user interfaces are possible. The authors with two others have developed an intelligent expert systems/laser disc prototype system under IST sponsorship which is an early attempt to link these relatively new technologies (Ragusa, et al. 1989). OSBATS, could also benefit from this linked system capability.

Satisfaction of Objectives

There is no doubt that OSBATS succeeds in arriving at a conclusion that aids users in the design and evaluation of training-device alternatives. These alternatives meet training requirements at the minimum cost or that provide the maximum training effectiveness at a given cost. The question is whether or not these objectives are achieved in an understandable, cost-effective, and appropriate manner.

For OSBATS to be a success, it must satisfy a need for the users, and the results must be clear to them. This is not necessarily the case. Although neither of the authors are Army engineers, both have spent many hours interacting with the system and studying the user manual.

The authors have found it difficult in some cases to fully interpret the conclusions and to compare/contrast different combinations of selections. There are two possible solutions to this confusion. First of all, the system could be modified to help analyze some of the decisions and to provide a comparison of different routes and the benefits achieved. An expert system front end would help to accomplish this. There is such a multitude of variables that it is difficult to ascertain exactly what the system is proposing. Another way to make the results more clear is to produce output data files or simple printed reports of the results of the analyses that could then be compared. Unless OSBATS can be made more straight-forward and understandable, users are not going to choose it as a feasible instrument.

The question of the cost-effectiveness of the model is beyond the scope of the investigators. However, once again, unless the users are comfortable with the model, they will not voluntarily choose to incorporate it in their concept formulation, logistics, or training development activities. Although instructional systems design models are advocated for all training decisions, they are often overlooked because of the time factor involved and because designers rely on their past experience.

OSBATS DESIGN ALTERNATIVES

Expert Systems Application

A logical question which could be asked is: Are there alternative methods for providing solutions to OSBATS objectives? Instead of an optimizing solution, perhaps an optimal approach to the problem is more appropriate. Two alternative approaches are suggested, one near term and the other which needs further investigation and time for technology and operational applications development. The first approach would be to consider a much greater use of knowledge-based expert systems technology discussed earlier for future OSBATS modification or redesign. The second approach, which is more futuristic, involves the integration of expert systems and neural network technology. Expert system applications are briefly addressed in the **Examples of Applicable Systems** section with a discussion of future hybrid integration of expert systems with neural networks in the **An Integrated Approach** section.

There are numerous problems that benefit from the application of expert systems technology. Some of these are management decision making, software development, intelligent computer-aided engineering, communications, command, and control, geophysical analysis, instrument interpretation, computer configuration, and intelligent computer aided instruction. Other applications include diagnostics, planning and scheduling, financial analysis, and other too numerous to list here. Solutions to most of these problems were not found by conventional means. Virtually any complex problem is going to benefit from this technology.

Man-machine interfaces, coupled with expert systems tools/methods, and the new programming environment make expert systems not only easy to use but more accessible. With recent advancements in both the development of expert systems environments and knowledge engineering tools, an experienced developer can capture expertise, put it into an expert system, and disseminate it to users. This environment creates an excellent means for solving complex problems more effectively and more efficiently.

OSBATS Evaluation

Is expert systems technology appropriate to the OSBATS problem solution? To answer this question it is necessary to discuss: (1) forms of expert systems applications, and (2) testing for applicability.

Forms Several forms of expert systems have been described in the literature and are generally described by category and problem type they address. Generic categories of expert systems applications are as follows (Waterman, 1986, p. 33):

- Interpretation — Inferring situation descriptions from sensor data.
- Prediction — Inferring likely consequences of given situations.
- Diagnosis — Inferring system malfunctions from observables.
- Design — Configuring objects under constraints.
- Planning — Designing actions.
- Monitoring — Comparing observations to expected outcomes.
- Debugging — Prescribing remedies for malfunctions.
- Repair — Executing plans to administer prescribed remedies.
- Instruction — Diagnosing, debugging, and repairing student behavior.

OSBATS falls into several application categories, however, two are probably most appropriate. They are predication and design. Prediction systems sometimes use simulation models (programs that mirror real-world activity), to generate situations or scenarios that could occur from particular input data. These potential situations, together with knowledge about the processes that

originated them, form the basis for the predictions. Expert systems used for design develop configurations of objects based on a set of problem constraints. Design systems often use synthesis to construct partial designs and simulation to verify or test design ideas (Waterman, 1986).

Testing for Applicability Numerous authors have identified tests to determine if expert systems technology is appropriate to a particular need. One notable evaluation test has been developed by Silverman (1989, p. 10) who identifies a rule set which describes when to "hire" an expert system. The rule-based format Silverman uses indicates that an expert system approach is appropriate if the solution is relevant, feasible, optimal, and success-oriented. More specifically the following criteria has been developed:

Relevancy Rule:

If:
1. There is a recurring shortage of skilled employees,or
2. Problems regularly arise requiring innumerable solutions to be considered, or
3. Job excellence requires unreasonably high levels of human training, or
4. No single person can know the requisite problem solving expertise, or
5. Difficulties in applying existing knowledge routinely cause management to work around basic problems.

Then: Expert systems are a relevant solution technique.

Feasibility Rule:

If:
1. The problem typically takes a few minutes to a few hours to solve, and
2. No controversy over problem domain rules exist, and
3. Problem domain experts exist, and
4. Problem domain knowledge can be cast into existing representational techniques.

Then: An expert systems approach appears feasible.

Optimality Rule:

If:
1. It is necessary to reason with erroneous data, ertainty, and make hundreds of thousands of judgments (heuristics), and
2. Inference engine planning, scheduling, and control procedures are more appropriate than conventional software program layouts, and
3. Interaction with human users via natural language is required, and
4. The computer program must be able to explain why it is asking a question, and
5. The computer program must be able to justify conclusions.

Then: An expert system is the optimal approach.

Success Rule:

If:
1. Solutions in the problem domain are of high value, and
2. Top management supports an expert system approach, and
3. An existing expert system kit can be used as that core of the applications, and
4. An iterative prototyping approach cam be pursued in which initial problems can be solved with a limited subset of the experts total knowledge, and
5. Training cases and test cases are available for gradually augmenting and evaluating the expert system, and

6. An apprenticeship approach can be adopted in which experts will review and critique each incremental version's solutions, and
7. The knowledge-engineering team has a successful track record.

Then: The expert system approach is likely to be a success.

From this criteria set it can be seen that there is sufficient justification for using a full expert systems implementation for OSBATS type applications. However, there are several questions which need further attention before a final determination can be reached. They are:

Relevancy -- Is there a recurring shortage of skilled employees?

Judging from the average experience level of the PM TRADE engineers (12 years), it could be assumed that this organization does not have turnover problems.

Feasibility -- Does a domain expert exist?

This question needs further assessment. Singer and Sticha (p. 30) indicate that OSBATS is an expert system based decision aid and that it does not model any single expert. Further, Singer in an interview, indicates that PM TRADE plans to conduct a survey in the future to determine if there is consistency in the way users perform their analysis. However, recent articles are reporting methods and results which allow multiple expert knowledge inputs.

Success -- Does top management support an expert systems approach?
Does the knowledge-engineering team have a successful track record?

The answer to this first question is not known. However, there must be some support in that OSBATS does use expert systems technology in part for its solution methodology. With regard to the second question, it is assumed that the knowledge-engineering team exists outside PM TRADE. Numerous organizations involved in expert systems development perform knowledge-engineering, including individuals at HumRRO who were responsible for the OSBATS knowledge bases developments.

Examples of Applicable Systems

According to Feigenbaum, et al. (1989) expert system shells are now commonly and successfully being used to provide a broad base of decision systems on a global basis. In fact, the authors report that after extensive study they have found more than 2,000 operational expert systems presently in existence. An appendix, contained in the Feigenbaum reference, describing expert systems in use has been compiled by Paul Harmon, a noted expert systems writer and seminar instructor. Each application is identified with a brief description of their environments and content. This report is too short for descriptions, however, several systems like OSBATS attempt to perform trade-off analysis between competing resource considerations. Most are from the business world and include credit and financial institutions.

The authors of this report have also been involved in development of prototype expert systems which reach solutions based on constraint and resource limitations (like OSBATS). Examples have been applications associated with determining manpower and dollar costs for NASA advanced launch vehicle configurations, and wage determination based on employee evaluations and available funding.

Design Considerations

There are numerous OSBATS expert systems design alternatives. However, two primary versions will be briefly discussed. The first establishes an expert system as the primary program with interfaces to support programs (e. g., data bases, spreadsheets, DOS calls,) accomplished through

hooks provided by the shell. This has been the most commonly used design for contemporary operational applications. Usually, interfacing is easy to accomplish with speed of interaction being directly related to CPU type and clock speed (286, 386, etc.), and how "well" the shell functions. As mentioned earlier, there are numerous commercially available shells which would serve as a design medium. The challenge is to find the one that best fits the application.

A second design recognizes that there are a number of existing computer systems and software in place. Consideration is given here to normal resistance to technology change and fear of the unknown (i.e. artificial intelligence and expert systems). More and more books and articles are indicating that the focus of this resistance is with MIS managers (Feigenbaum, 1989). To alleviate this concern, several organizations are attempting to use existing main frames and non-AI languages instead of LISP and PROLOG--the primary AI development languages. A popular high-level language used is C. A commonly used technique is to use a shell to build the knowledge base, and then port the resulting program to C. Since data base capabilities can exist in C as well, an integrated software system can reside on the computer system of choice.

Some shell developers such as the NASA Johnson Space Center's Mission Planning and Analysis Division have implemented an expert system shell in C. CLIPS is a tool which was designed to allow expert system construction and delivery on conventional computers. The primary design goals were portability, efficiency, and functionality. To meet these goals CLIPS is written in, and fully integrated with, the C language. An Ada version is about to be released.

An Integrated Alternative

Recently neural networking concepts and technology have challenged expert systems as the wave of the future. Numerous articles are appearing in the popular AI technology publications and in academic journals. While neural network concepts are still emerging, research and applications are being reported. Briefly, neural networks (or neural nets for short) are used to look at patterns in a set of sample cases--called a training set. Neural nets are then used to learn these patterns, develop the ability to correctly classify new cases based on the patterns, and then correctly classify new cases based on the patterns learned by the network (Sherald, 1989).

As has been discussed earlier, expert systems are best if a problem exists that can be easily described by a set of rules. In general, neural nets are better at problem diagnosis, decision making, and other classifying problems where fuzzy logic or pattern recognition is important. There are several other specific instances where neural nets have an advantage over expert systems according to Sherald (1989, p. 10). The two situations are: (1) where you have non-procedural tasks where it is known how to solve the problem but the rules are not known, and (2) where the problem is procedural but it is too expensive to capture all of the rules e.g., patient diagnosis. One problem with neural nets is that they presently do not (like OSBATS) have explanation capability. Some researchers estimate that perhaps such a capability will exist in five years.

Several researchers are predicting that both technologies will merge to take advantage of the strengths of each and the synergy which might develop. For example Dr. Robert Hecht-Nielson sees the marriage complementing each other rather than one eliminating the other. "Neural networks can process raw data and provide conditioned output at a higher cognitive level, e.g., in category determination. The output is given to an expert system for further processing."

RECOMMENDATIONS

Short Term

While not a complete list, the following are some suggested changes to the existing OSBATS program which might help in the short term (less than one year).

General

Expert Systems -- Incorporate an expert system front end to help users determine mode of operation, input responses, analyze decisions and make evaluations.

Training -- Provide training for the system either in the form of lecture/practice or on-line tutorial/simulation. Perhaps a 3-5 day course would suffice.

Service Contractor -- Because of the complexity of OSBATS, consider using a service contractor in a support mode to provide OSBATS analysis for PM TRADE users. The obvious danger is the loss of analysis capabilities and diluted responsibilities of civil service personnel.

System Design

Run Time -- Provide closer approximation of the length of time required to run the resource allocation module. A time of 2-20 minutes is a bit ambiguous. Also it should be made clear if this must be run for each allocation problem or on a one-time basis.

Scenarios -- Provide scenarios for the user to practice with the system after going through the examples in the manual.

Use of Keyboard -- Inputs requiring the keyboard should be minimized.

Shortcuts -- Shortcuts should be provided for experienced users.

Dictionary -- On-line dictionary of terms should be provided.

Asterisks -- Printouts containing asterisks are confusing (on the screen they are color-coded). Perhaps a different character could be used for the different categories.

Title Consistency -- Provide consistency in titles for menus and screens.

Name Consistency -- Provide consistency in names of branches.

Highlights -- Provide highlights in help screens.

Terminology -- Change terminology on Delete/Save Options screen to read Ignore/Save Options.

Minor Corrections -- Correct grammatical and spelling errors.

User's Manual

Reports -- Provide printout of report and comment capability without requiring the user to exit to DOS.

Continuity -- Although the users manual is very helpful, there are some modifications that could be made. It is sometimes unclear what action the reader is to take, often resulting in the necessity to backtrack after choosing the wrong route.

Flowcharts -- Incorporate user manual flowcharts of the system structure.

System Bugs -- There are some "bugs" in the system that cause a TERMINATION ERROR to occur when using the manual (especially on page 71 of the manual under "Including Instructional Features").

Minor Corrections -- Like the User's Manual grammatical, spelling errors in the system should be corrected.

Future The following are several suggestions regarding OSBATS which should be considered. Implementation would take longer than one year and are therefore are placed in this category.

OSBATS needs to be a generic shell (i. e., appropriate for a wide variety of military applications) with a standard interface, fast action response, and an explanation capability. The authors agree with Singer and Sticha report that advocates "expanding the prototype OSBATS from the current rotary-wing operations domain to other domains" (p. 30). This would make the system more cost-effective and allow a wider range of implementation.

Specific needs for a generic system are as follows: (1) domain specific knowledge (dynamic), and (2) domain independent knowledge (static). The domain specific part needs to be:

- Easily modified (e.g. rules, data base).
- Domain expert (single or consensus based) knowledge based.
- Portable to other computer systems (e.g. workstations).

The static part needs to be:

- Broadly validated.
- Relatively easy to update.
- Rich in universally accepted concepts.
- Domain expert (single or consensus based) knowledge based.

There appear to be at least four future options available to PM TRADE relative to OSBATS in its present form. They include but are not limited to the following options. Variations are of course possible. Advantages and disadvantages are listed for each.

Option 1 (Abandon) -- Consider the present OSBATS prototype as a system which should be abandoned.

Advantages:

- No further funding is required.

Disadvantages:

- OSBATS funding did not produce an operational system.
- A decision aid is still not available for users.

Option 2 (Rebirth) -- Throw away the present prototype and begin again with the same approach (optimization model) working toward a generic, more user-friendly, integrated, and operational system.

Advantages:

- Earlier investment will not be totally wasted.
- Generic system could result if successful.
- Should have a better chance of user acceptance.
- Development team is still in place.

Disadvantages:

- System still based on theoretical (optimization) model.
- Could result in another black box system.
- Will be costly to develop.
- May still have user resistance.
- Could require a lengthy development period.

Option 3 (Evolution) -- Evolve the present optimization prototype into an operational system working toward a generic and more user-friendly system. Ensure a totally integrated design with better use of expert systems technology.

Advantages:

- Would take maximum advantage of earlier OSBATS investment and development with little wasted.
- Could have an operational system in a shorter development period than Option 2.
- Would be less costly to develop than Option 2.
- Would use selected expert systems technology.
- OSBATS developers have most of the skills needed.

Disadvantages:

- System would still be based on theoretical (optimization) model.
- Could result in another black box system.
- May still have user resistance.
- Could still be a costly development.
- System could be slow because of extent of program interaction.

Option 4 (Redesign) -- Accept an expert systems (satisficing) approach to redesign the entire system using all rules and data that can be saved. This approach would build a rapid prototype to demonstrate proof of concept before an operational system is built.

Advantages:

- Would take advantage of earlier OSBATS investment and development with little wasted.
- System should run faster.
- Should be a cheaper development than Option 3.
- Has a better chance of user acceptance.
- Would contain explanation capability and help manuals, eliminating black box image.

Disadvantages:

- Could run into new design problems.
- OSBATS developers may not have necessary expert system development skills.
- Knowledge engineering team is needed.

Recommendation

A review of the options presented can lead to different conclusions depending on several unknown variables. The rule-based possibilities are:

Rule 1:

If: 1. PM TRADE feels that the need for an OSBATS-like decision aid does not exist, or
2. Additional funding is not available.

Then: Accept Option 1 (Abandon).

Rule 2:

If: 1. A need still exists, and
2. Sufficient funding exists to continue this effort.

Then: Perform a trade-off study between Options 2 (Rebirth), 3 (Evolution), and 4 (Redesign).

Rule 3:

If: 1. A need still exists, and
2. Sufficient funding exists to continue this effort, and
3. An optimization solution is required.

Then: Select between Options 2 (Rebirth) and 3 (Evolution).

Rule 4:

If: 1. A need still exists, and
2. Sufficient funding exists to continue this effort, and
3. Satisficing solutions are acceptable, and
4. PM TRADE is willing to support an expert systems approach, and
5. A knowledge engineering team exists, or
6. A knowledge engineering team can be contracted.

Then: Select Option 4 (Redesign).

As can be seen, a finite set of decision possibilities can be identified depending on the condition of the variables which have been listed. Since the authors do not know what conditions exist, it is

difficult to make a final recommendation with certainty. However, if certain assumptions are made, it is possible to focus on one option. If the following assumptions are true, then Option 4 is recommended:

- A need for an OSBATS decision aid to provide training device decision consistency remains.
- Additional funding can be obtained because of the potential for significant overall training program cost savings.
- Satisficing solutions are acceptable.
- PM TRADE management will support an expert system.

Under these conditions Option 4, a redesign of OSBATS as a satisficing expert system, is suggested. With this approach, the prototype development of the present OSBATS system would be used as the foundation for a redesigned system. Rapid prototyping would require one or more PM TRADE identified user experts. From this effort would evolve a proof of concept system which would demonstrate sufficient depth and breath for the AH-1 domain. User and PM TRADE review, validation by expert case testing, and acceptance would follow prototype building. A phased development approach would then be used to construct an operational AH-1 system followed by an evolution to a more generic system. However, before beginning the rapid prototyping stage, it would be understood that domain specific (AH-1) and generic (weapon systems) components of the system would be identified and initially developed separately and then integrated.

While this is an over-simplified description of options, variables, assumptions and recommendation possibilities, it is not without basis. Numerous operational systems have been built or redesigned from felt needs or existing systems which did not fully satisfy their initial objectives. Future study will be required to answer several important questions raised in this report. They include:

- Is PM TRADE willing to continue development of automated training decision-aid system?
- What would an evolved or redesigned OSBATS-like system cost?
- Is there a need for a generic system?
- Which parts of the existing OSBATS are generic and which are domain specific?
- Are additional funds available for further developments?

SUMMARY AND CONCLUSIONS

Summary

Results of this study should be of value to OSBATS sponsors and others for a variety of purposes: (1) better understanding of the existing system from a different perspective, (2) specific comments concerning the advantages and limitations of the present design, (3) presentation of design alternatives (namely a partial or full expert systems approach), and (4) some short and longer term recommendations for the future.

Conclusion

The reasons for funding the existing OSBATS system prototype are still valid, since there can be substantial savings for training device development when the high cost of weapon systems acquisition and operator training is considered. Too often seat-of-the pants analysis is done because of the lack of criteria or standards. In the opinion of the authors, there are significant cost savings and favorable cost/benefit advantages which could be realized through the development and use of automated decision aids. While this cursory report may not provide total and complete answers, perhaps it does provide some visibility to the next step.

SECTION II
USER'S EVALUATION OF TWO TRAINING SYSTEM DESIGN AIDS: OSBATS and ASTAR

Steven Gibbons
Institute for Simulation and Training

This section is divided into separate discussions on OSBATS and ASTAR. The OSBATS discussion describes the purpose, installation, and operation of the design aid from the viewpoint of a user. The feasibility of expanding the use of OSBATS to a new domain (tank gunnery) is also explored. The ASTAR discussion is a user's analysis.

SECTION II

USER'S EVALUATION OF TWO TRAINING SYSTEM DESIGN AIDS:

OSBATS and ASTAR

(OSBATS) OPTIMIZATION OF SIMULATION BASED TRAINING SYSTEMS

This section describes the purpose of OSBATS and critiques the installation and operating procedures from a user's viewpoint. Factors discussed include quality of model solutions, screen display, model reasoning, and documentation. The functions evaluated were taken from the user's guide (Gilligan, Elder and Sticha, 1988).

Purpose of OSBATS

The overall purpose of OSBATS is to determine training device and training system configurations that meet training requirements at minimum costs. Training requirements relate to the specific level of student performance expected after training. OSBATS examines only those tasks that can be trained by a training device or simulator. OSBATS is designed to perform three tasks:

- identify tasks that are good candidates for training on a training device
- design training devices with a level of sophistication and cost tailored to the training requirements
- allocate training resources among training devices and actual equipment to minimize cost.

A data base for one domain has been developed for OSBATS (rotary winged aircraft) to allow user manipulations of all module functions. This prepared data base is designed as a method of helping the users to understand the operations and use of the system. The user's guide is designed to illustrate both process and procedures of operation. The guide is intended to complement standard reference manuals. Both the subset of modules that are used and the order in which they are used may vary depending on the requirements of the problem and the preferences of the users. This means that large variations in the method of applying the model exists. The many different methods of implementing the model makes critically evaluating any one method difficult.

General Comments

OSBATS is a data intensive approach for training device standardization. (See Section I for an overview of OSBATS.) It requires specific information on domain parameters (resident data), and assumes that design trade-off decisions are different between training domains. The model derives cost estimates based on the designs fidelity configuration and historical cost data. The model structures design trade-off decisions based on situational specific task (non-resident) data. The model is designed to be used by experts and uses expert terminology in describing the model and its outputs.

An expert system rule base organizes design questions in the same manner that an expert might. The outputs of the system include cost and benefit values for levels of fidelity and different instructional features. It is capable of designing the entire training program and informs the user which TD to use and how many are needed.

OSBATS is still in prototype development and has not been fully validated. Presently, the model is constructed to handle only training device designs in the rotary wing domain. The validation of internal concepts and dependencies assumed by the model has not been completed. It is presently one of the most complex computer based methods available to assist training device designers in their trade-off decisions.

OSBATS' major deficit is the lack of appropriate and complete documentation. The effective use of the model relies on a creative synthesis of the presented information. Each module offers different information for differing portions of designing a training device. The documentation provided does not explain the range of possibilities in the models usefulness. Documentation does not outline the number of choices available within each module and where finally decision points are made. Assumptions made in early modules can not be easily traced to their consequences.

OSBATS makes numerous computations within each module that are used as system inputs for latter modules. These calculations are made using variables, which require choices to be made by the operator. Once a module's results are computed and saved, only these totaled values are used as values for other modules. This method of operation eases system computational needs. Difficulties arise when users change input values after module computations have been completed. Variables used to calculate values for one module can be changed in a second module without corrections made to the first modules results.

The legends at the bottom of the display can be confusing at times. All available options are not continually displayed. In some modules, certain menu options can only be reached while within a specific display. The user can never be sure that all menu options have been searched. This hierarchal problem is magnified because of the modular construction of the selection menus. Once inside a module, the ordering of selection is vague. The impression of non-unity is conveyed since the upper level options are easily distinguished, while lower level selections are not all displayed.

During familiarization of the model, the instructional conventions of the users manual changes. As users progress further into the manual, descriptions and explanations continue to get less concise. This is especially true in the Training Device Selection Module. The first module (Simulation Configuration), does not contain a module introduction paragraph. All other modules begin with a short introduction, explanation and prerequisite inputs required. In the RESAM (the last module), one main menu option, Save Module Constraints, was not addressed in the user's guide.

The modules do not make clear where the user should stop examination and make system choices. The goal of each module is not clear. In the IFSM, the last display (cumulative B/C graph) in a series of informational graphs is the point in the module where the main module decision must be

made. The operator must choose an instructional feature package which is the major output of the IFSM. Yet, by embedding this screen deeply in the module structure its relative importance is hard to grasp.

When the model is being started, one of the first tasks is to load the data base by selecting a task cluster from the library. The system begins calculations while the user waits. The impression is given that the specific task cluster chosen will be loaded. This is not the case; the default task set for any cluster chosen will be loaded. When the user returns to the Selection library to chose a second task cluster to examine, it is expected that the system will reload another data set, requiring system loading time. However, the data base is immediately loaded and the model is prepared to operate.

For further detail about module functioning, refer to appendix A. Appendix A takes a descriptive approach to all module functions in a chronological sequence.

Installation

Hardware Verification Before the model can be used, it must be ported to the host computer. The porting of the system requires that the host computer be configured correctly. The OSBATS software requires a IBM PC/AT, Zenith 248 or compatible, with 640K of memory, and a ten megabyte hard disk. In addition, an enhanced graphics adapter (EGA) and monitor, a 80287 numeric co-processor, and a Microsoft compatible mouse with mouse driver installed are required. (Real-time interactions are not possible without the co-processor.) The **PRINT.COM** file must be in a directory specified by a path command.

The OSBATS software is not compatible with the **ansi.sys** driver found in many personal computers. Thus, the user must remove this from the **config.sys** file and restart the system. The **ansi.sys** file has no adverse effects on the computations of the model, but begins to cause problems as screen displays become more detailed. For example, help screens that are over one page in length cannot be completely read because the screen scrolls past the first few lines and goes directly to the end of the explanation.

The installation guide does mention this phenomena and gives these basic instructions for correcting the problem:

> To determine if the ANSI.SYS driver is on in your system examine the file called CONFIG.SYS. If the file listing includes the line DEVICE=ANSI.SYS then you must use a text editor to remove this line from the file. Save the edited file and reboot the system.

A novice user may not be able to understand these instructions. The importance of correcting this problem at the start of model installation is not properly emphasized. If it is not corrected at this early stage, problems arising during operation will be difficult to trace back to this point. No examples of system consequences are provided in the guide.

A more simplistic approach may be helpful at this early stage of user interface. If OSBATS uses an expert system to run many of its functions, perhaps a system check for the user could be included. Verification of the hardware system is a procedural task that can be removed from the user's responsibility. For example, once OSBATS is engaged, the first check for the system could verify that the host environment is congenial to its functions. Any discrepancies found could either be corrected automatically or brought to the user's attention, along with detailed guidance of how to correct the problem.

Another problem involves the mouse required by OSBATS. (A mouse is a small mobile input device that controls movement of the cursor on a computer display). The mouse is needed for most interactions with the system. Reliance on an input device that a user may not be familiar with can hinder user acceptance. Unless a specific need exists to have a mouse input device, the user should have the option to use the mouse device or the pointer keys present on keyboards.

Software Installation Detailed instructions are provided in the guide for installing the software. Procedures for installation are divided between installation of the data files and of the program files. If OSBATS is being installed on the system for the first time, installation of the program files and the data files are required. If the OSBATS model is already installed, the new data files need only to be reinstalled. The reinstallation will erase existing files if a earlier version of the software is present.

OSBATS is delivered with two sets of data, one set contains 125 tasks found in AH-1 operation, and the other set contains 38 subset tasks from the AH-1 domain. These task lists have been developed specifically for the OSBATS model. All tasks are described in OSBATS model data requirement formats. This prepared database is designed as a method to help users understand the operations and use of the system. Since the data is already prepared, the user is alleviated from having to collect and encode data.

The software installation procedure consists of 19 steps, five of which are for file verification. The procedures are explained clearly enough for a new computer user to be able to complete the installation unaided. However, there is some variation in the format that may cause confusion at two points.

First, instructions for installing the first two program disks are clear and follow the same format, but instructions for the next disk are listed only as repeat this step. This guidance may be confusing since it lacks the detail given for all phases. Confusion might be cleared if the repeat command were given an installation procedure number (as the others had been). Presently, this command is embedded in the installation procedures of the Program Disk 2 instructions.

The second point of confusion is in the Installation of Data Files instructions. In step 14, the installation instructions displayed by the computer are listed in the user's guide. Instead of giving the response required (as with other steps), a paragraph is listed explaining that the user should input an <A> "if you want to install data for all AH-1 tasks, or <S> "if you want to install the data for the sample problem in the user's guide." (Option <A> should be chosen only if the user is familiar with the software. The user can reinstall the <S> option.) These instructions do not make clear which option to choose. A careful reading of the guide is necessary.

Operation

Start up Procedures After successful installation and verification, the following set of procedures must be performed each time the system is started.

Specific DOS instructions are provided for reaching the appropriate subdirectory. The OSBATS title screen appears listing program identification information. At this point, the user's guide adapts a presentation format in which responses required by the operator are indented and labeled **ACTION**. This format quickly alerts the reader that some interaction with the program is required. A left button click on the mouse begins the models operations.

The first system choice screen comes up. This screen presents the user with the option for recalculating the data. An unclear note explains why recalculation would be required. The note mentions the need to first interact with the rule base if appropriate instructional features for the AH-1 tasks data is available. The guide's **ACTION** instructions clearly state that this option should not be chosen. Instead, the option to run OSBATS using current values should be chosen.

The next screen immediately presents the user with a second choice similar to the first. The user must decide whether to recalculate the data using the fidelity optimization features rule base. A short unclear sentence explains why recalculation might be needed. The guide's **ACTION** instructions clearly state that this recalculation option should not be chosen. Instead, the option to run OSBATS using current values should be chosen.

Next, the task cluster library is presented. The task cluster library is used to select a cluster of tasks for demonstrating OSBATS features. The task cluster labeled default contains all tasks included in the data base. All other task clusters are subsets of the default task set. The first time the system is started, the only choice available is the default task.

A few problems exist with this screen presentation. The documentation deviates from format standards. The screen lists an **ACTION** where no interface action is required. Also, the screen instructs the user to SELECT a cluster option but the guide does not identify this interface as an ACTION. The manual presents a block insert of how the Task Cluster Library should appear, but this insert is different then what a first time user will encounter. The insert contains subclusters that can only be produced through module interaction. Since no interaction has taken place, no subclusters should be present. The manual and OSBATS refer to this menu using three different labels. The guide first labels this menu The Task Cluster Selection Library, then refers to it as Task Cluster Library; OSBATS main menu labels it Task Cluster Selection Menu.

When subclusters are present, an option to delete them exists. The deletion is accomplished by selecting the subcluster with a red deletion strip. This deletion strip is located next to the yellow strip which indicates selection. The subclusters are very close together, making it somewhat difficult to select the desired subcluster. Some additional spacing between subclusters would help alleviate this problem.

Another selection problem is present for the two preceding recalculation choices. The options to recalculate or run model are only one line apart. Fine movement control of the mouse is not possible, thus cuing errors are highly possible.

Finally, instructions about editing a task cluster are provided in the guide at this point, yet no method of editing a task cluster exists this soon in OSBATS. This can only be done within a main menu of selected modules.

Once a task cluster has been selected, the rule base begins to generate the data needed by the system. A short wait is required (one to two minutes). OSBATS presents a screen that tracks calculation progress. This screen uses a blue thermometer to gauge the amount of distance to the end of the calculations. This screen is clear and very helpful. If the model stops calculations or begins a continuing loop, the thermometer will stop.

If the host system is not equipped with the 80287 numeric coprocessor stated by hardware requirements, the thermometer will stop short of its completion position. The thermometer starts quickly and stops at about one fourth the distance it should travel. Once stopped, the bar does not continue. This is a result of the numerous calculations that must be made prior to system interaction. The problem arises in that a host environment not containing the numeric coprocessor will still attempt to make all the calculations. This requires time in excess of 45 minutes. OSBATS does not inform the user of the wait; the instructions appear as if the next screen will appear automatically. The user may be unaware that a problem exists and may wait for an extended period. Even when the user realizes there is a waiting period, it is not clear how long the wait will be. Since the user is not informed that a short wait is required, the user may not be surprised if calculations require over an hour before the system starts.

It becomes helpful once again to provide an automatic system configuration check at this early stage. If the numeric coprocessor is not present, the system should immediately stop processing and notify the user that a systems requirement has been violated. Failure to notify the user results in the user expecting the model to run correctly without fully understanding why the system is moving slowly. In the first running of OSBATS, the host computer did not contain the numeric processor required, and initial calculations exceeded 45 minutes and were still not complete.

Overall, the procedures for installing software are simple and directions are clearly presented. New computer users can reasonably be expected to port the system to a properly configured host computer. Total installation including verification of all files and familiarization of model convention took one and a half hours to complete. At this time OSBATS is fully functioning on the host computer and further excursions into its depths are possible.

Software conventions General procedures for the operation of the software and display conventions are described in the guide before operational description of the modules begin. These conventions appear in all modules and include:

- yellow background areas indicate active fields
- clicking mouse in yellow area control entry
- yellow with black numbers indicate available option
- black with yellow numbers indicate unavailable for use
- red indicates item is subject to deletion or temporary elimination
- left button on mouse is used for model interaction
- all keyboard entries must be followed by <enter>
- the screen is divided into three sections:
    - top five lines are blue and contain model status information, current task cluster, and status of weight values
    - bottom seven lines are blue and contain menu options available to the user
    - middle 31 lines present graphs and tables
- PRINT SCREEN, prints all but the last seven lines
- SAVE SCREEN, option works only in the TDSM
- Toggle screen, compares the saved screen with present screen

OSBATS has two system conventions that allow the user to record interaction with the system. The first feature allows the user to record notes as OSBATS is running. The second feature allows the user to record all screens viewed and the number of comments made during the run.

In the first feature, comments are saved starting at 1 each time the model is begun. This means that all comments made in the last run of OSBATS must be renamed or they will be written over by any new comments. Also, review of comments is not possible during system operation; once a comment is saved, the model must be exited to examine it. Combined, these two problems render the comment feature nearly useless.

The second feature contains an ordered list of each OSBATS display that was viewed and the number of comments made while viewing this screen. This function updates with each model run. In order to retain this information, the file must be renamed before the next system run.

The guide does not explain the temporary nature of these features. Information saved may be lost before the user discovers this system flaw.

Module Use

OSBATS is now ready to begin the user interaction process. After the blue thermometer indicates the rule base is loaded, the Module Selection Menu is presented. Any of the five modules can now be examined. Modules may be accessed in any order, depending on the needs of the user. However, the Resource Allocation Module (RAM) is only meaningful after a training device package has been developed from the Training Device Selection Module (TDSM). For the purpose of this report, all modules will be reviewed in sequence of main menu ordering, since this is the logical progression of modules during the designing of a training device. The modules follow this order:

- Simulation Configuration Module (SCM)
- Instructional Feature Selection Module (IFSM)
- Fidelity Optimization Module (FOM)
- Training Device Selection Module (TDSM)
- Resource Allocation Module (RESAM).

The Simulation Configuration Module addresses task selection, the Instructional Feature Selection and Fidelity Optimization Modules address sophistication and cost trade-offs, and the Training Device Selection and Resource Allocation Modules addresses the allocation of training resources.

Simulation Configuration Module The Simulation Configuration Module (SCM) classifies all training tasks into three clusters:

- tasks to be trained on a full mission simulator (FMS)
- tasks to be trained on a part mission simulator (PMS)
- tasks to be trained on the actual equipment (AE).

The SCM examines each task to determine the need for simulation and the cost-effectiveness of using a simulator to train rather then the actual equipment. The segregation of tasks is based upon task training requirements and the potential operating cost savings obtained with simulation training.

This list outlines the specific inconsistencies and difficulties a user will encounter when operating the SCM. A few of the comments and questions are answered in OSBATS documentation. Comments on the following modules have identical formats.

- The reasons for manipulating the Training Requirements Index (TRI) and the Operating Cost Boundary (OCS) are not adequately explained.

- A dependency exists between the TRI and the OCS values; as one value increases the other decreases. What is the basis for the dependencies?

- Allocation of a cost savings value to each task (which is needed to obtain a indication for simulation value) is provided before any module operations have begun. This is confusing since no clear explanation of the origin of these task values is provided.

- The Cost Boundary Slope (CBS) values range between 40% below baseline to 40% above baseline (the values increase and decrease in steps of 20%). Baseline is not described adequately. No detailed explanation is given as to where the baseline figure is calculated from or why increments of 20% were used.

- The implications, effects or consequences of altering the TRI, OCS, CBS or the Task Complexity Line (TCL) on later model operations is not described. Do these values have any other relevance other then sectioning the task cluster?

- The development cost of the simulator is determined from task cue and response requirements, yet no information is provided of how cue and response requirements are allocated to tasks.

- The indication for simulation value (the main graph's x-axis) is a vague value that has had none of its calculations or specific origins outlined. A brief definition is supplied but it is not detailed enough to give the user confidence in the values.

- The first display (Graphical Summary) contains too much information and gives impression that the system is complex.

- The amount of information presented in the Graphical Summary display is useful to an expert model user, but a new user attempting to gain model understanding might not be able to assimilate the large array of information.

- The origin of the fidelity dimension requirement values for training a task is not provided. The data is presented in a manner that implies the model has calculated the values. However, this information is imputed by the user during the data collection and encoding. If the user is not the same individual who constructed the data base then confidence in these values will be lessened.

- The Sensitivity Analysis screen information is difficult to understand.

- The y-axis of the Sensitivity Analysis graphs are not labeled.

- When saving the newly developed task clusters, the user is not allowed to name the grouping. Labels are determined by the model.

Instructional Feature Selection Module The Instructional Feature Selection Module (IFSM) begins the training device design process. Instructional features are elements of training devices that can improve training efficiency on individual tasks. This module requires a task cluster obtained through the SCM or defined by the designer. The IFSM uses a set of rules to compute benefit values for each instructional features on a task by task basis. OSBATS presently contains 21 candidate instructional features. The development costs are matched with individual instructional features to obtain a cost-benefit values. OSBATS then orders the instructional features in descending benefit to development cost ratio. The IFSM then requires the user to select a group of instructional features for further examination in Fidelity Optimization Module (FOM).

The IFSM begins by selecting or creating the task cluster. Selection is done through the Task Cluster Library described earlier. After selection and return to the model main menu, the user should select the Instructional Features option. If no appropriate task clusters exist, the user can enter directly into the IFSM.

This list outlines the specific inconsistencies and difficulties a user will encounter when operating the IFSM. A few of the comments and questions are answered in OSBATS documentation.

- The instructions are not clear as to what happens to instructional feature set changes. The changed grouping does not appear as a selection choice, but is embedded in the task cluster that is labeled and saved by the user.

- The saving of altered task packages waits until all manipulations have been completed. The reasons for not immediately saving newly developed lists are not supplied, nor are adequate directions given as to why the Examine/Edit options might be implemented.

- If task clusters previously defined with the SCM are changed at this point, assumptions made in the SCM are invalidated. This is not explained in the documentation. A set of user cautions statements is needed to inform the user of the full impact of editing task clusters.

- Editing task clusters early in the model violates only a few working assumptions; however, as deeper levels of the model are reached, changing a task set will result in greater consequences.

- The first screen encountered in the Display Results option is the summary display. Within this display, task weights and feature weights can be excluded or included in the calculations. Instructions explaining the consequences of including or the excluding of weights are not provided.

- When task weights are included in the calculations of total benefit, task weights are handled as decimals, and when excluded, task weights are handled as whole numbers. This discrepancy does not change the calculations since the values are eventually normalized. It only becomes a problem when users attempt to compare the difference between excluding and including task weights.

- The screen layout of task and feature weights is reversed. Changes in task weight inclusion, located on the right side of the screen, result in changes of column values located in the left side of the screen. Instructional feature weights are included on the left side of the screen while column changes appear on the right side of the screen. This screen variation makes it difficult to see what values have changed when weight are excluded.

- Inconsistencies in labeling also exist. The screen labeled Summary Display is called up by selecting a menu option titled Results Table.

- When changes occur in the values assigned to the Instructional features, the order of the colors in the modules graphs (each instructional feature is represented by a different color bar) stay the same but the IFs change position on the X axis and in the legend. This makes comparisons between changes more difficult.

- The colors in the graphs are each repeated three times when all 21 instructional features are present. There are more IFs then available colors to graph them.

- The y-axis is not labeled on any of the graphs.

- The fourth graph titled Cumulative B/C Graph is the most critical of all screens within this module. Within this screen "ifout" packages are saved for use in the FOM. The FOM requires a created ifout for module functioning. This information is not stressed strongly enough. This graph is embedded in the fourth level of a sub-menu option.

- During this module one major system convention has been violated. When the Assignments List menu option is chosen, the sub-menu choices appear in a field designated as presentation areas of graphs and tables, not designated menu options areas.

- All module options are not displayed continually; when an option is chosen, the option label disappears and is replaced by the title option of the previous display. This results in menu options switching positions throughout the analysis.

- Presentation of a Save Changes option in the main menu implies that the operation is required. This option does not always need to be exercised. When changes have been made that require saving, the model will automatically place the user into the Save Changes screen when he attempts to leave the module.

Fidelity Optimization Module The Fidelity Optimization Module (FOM) continues the training device design process begun with the IFSM. The FOM has two performance goals. The first goal of the FOM is to specify the levels of technical sophistication on a set of fidelity dimensions. OSBATS presently contains 11 fidelity dimensions varying in the number of levels. Each dimension level has an associated benefit and development cost value. The FOM selects dimension levels to maximize the effectiveness of a training device given development cost. The FOM may recommend several different device configurations at different device costs. The second goal of the FOM is to make trade-offs between technical sophistication of presenting cue and response feedback and technical sophistication of instructional support. This requires the use of the prioritized list of instructional features developed in the IFSM as input.

Before operation of this module, two inputs must already exist: the cluster of tasks created in the SCM and a saved Instructional Feature package. The IFSM must first be used to develop a design package of instructional features. These design packages should have been saved using unique titles.

This list outlines the specific inconsistencies and difficulties a user will encounter when operating the IFSM. A few of the comments and questions are answered in OSBATS documentation.

- Only one IF package can be examined at a time with a given task cluster. A comparison between Instructional Feature packages is not possible.
- Examining an ifout package after it has been created is not possible.
- When a task cluster is chosen that has not had the IFSM operation performed on it, no option to Include Instructional Feature Results is presented in the main menu. Yet, all other main menu options are still functioning.
- There is little spacing between the displayed listings of ifout packages, causing possible mistakes in cuing. This cuing problem is compounded by the fact that none of the following screens inform the user which ifout package is being used.
- If the user changes the ifout package, it is unclear if the first package is replaced or if the new package has somehow been incorporated into the first package.
- Problems arise when the system allows users to interact and change a task list. The benefit values within the IFSM were used to select the IF packages the FOM depends upon. If the task cluster is changed, values change and previous choices made may become invalid. If changes are made in the task set, then changes must be made to the IF packages.
- When tasks weights are excluded, documentation is unclear as to what happens. The manual states "...if the goal of a training device is to provide familiarization training on a variety of tasks, the task weights... may not be appropriate." How this translates into the analysis is vague.
- When task weights are excluded, more points appear on the optimal device curve. The reason for this increase in optimal devices is unclear.
- Is the assumption being made that for all tasks and training devices that the fidelity dimensions remain in the same order of importance? Does the expert system account for the possible differences of fidelity dimension ordering for different training devices of the same set of tasks?

- In specifying cost values, OSBATS lacks precision. The value of OSBATS is its cost prediction capability. Empirical data is used to obtain the high and low values, but each point between is interpolated using technical performance equations.

- Cost figures for different fidelity levels are static in that they do not change between device configurations. Is there not a dynamic relationship between dimensions and levels of all the features? Do these cost represent this changing relationships between levels of dimensions?

- In the third matrix table, benefit values for a given level of any fidelity dimension are given. The values range from 0.00 to 15.70. What is the scale value? What does a value stand for? Higher numbers do indicate level of importance, but why a fidelity level has a value of .09 is not explained in the guide.

- An inconsistency appears when choosing a specific matrix table. Once an option is chosen, the fields should switch to yellow numbers on a black field, indicating a option in use. This does not occur. The yellow option block stays yellow after selection.

- The matrix display is the first point where the Print option is usable, yet no explanation is given as to why or how the user would use it.

- No guidance as to what order to view the screens is given or implied. No indication as to which display is the most critical is given.

- No where does it state that the directive of the FOM is to select varying points on the B/C curve (optimal training device design) for evaluation and later model use.

- The module allows the user to edit the task cluster. This editing invalidates the IFSM and the ifout being used.

- Much of the information that drives this module comes from task definitions. The method in which tasks are defined is critical to accurate system functions. Yet, no literature exists on how to code a task list that is meaningful to the system.

- How detailed is this taxonomy? It is the strength of device fidelity configuration decisions, and is the high cost prediction driver. Yet little explanation is presented to the user as to where the dimensions and levels came from or why each is significant, or if any validity is present. The matrix tables present cost and benefit values but do not convince the user that they are accurate.

Training Device Selection Module In the Training Device Selection Module (TDSM), the focus of OSBATS shifts from designing training devices to integrating individual training devices into a whole training system. This integration includes an analysis of existing devices, classroom training, and the actual equipment as parameters in system design. The user's guide does not adequately prepare the user for the change of focus. The module appears to be continuing the design of optimal training device when it has already accomplished this task. The documentation could make it clear that if individual training device design is the goal of the user, then model operations are complete. The training devices produced in FOM are designed as precisely as the model is capable. Further model analysis focuses on system designs.

The TDSM integrates individual training devices into a training system. A number of devices, actual equipment, classroom training, existing devices and prototype devices are combined into a training system. The TDSM applies cost-effectiveness analysis to select the training device that meets training requirements for all tasks at the minimum cost. The TDSM assumes that cost is a simple linear function of training device usage. The module then allocates tasks to training

devices within the system based on the device that produces the best level of performance given the cost. The TDSM minimizes the cost to train each task on available training devices using learning curve data. The output of the TDSM is the aggregate use of each training device across all tasks. The TDSM recommends the device that should be used to train each task, the level of training at which each device should be employed, projected life-cycle cost of training, and the expected performance level of each selected device.

The TDSM is designed to inform the user of how many devices need to be purchased based on a training system utilization plan. Before operation of the module, three inputs must exist: a created PMS or FMS task cluster, an ifout package, and a set of new training device designs. The default task cluster must be in use by the system.

This list outlines the specific inconsistencies and difficulties a user will encounter when operating the TDSM. A few of the comments and questions are answered in OSBATS documentation.

- It is not clear why tasks would need to be edited since the training device has already been designed.

- The overall summary table could be restructured to convey more effectively the importance of the values. The cost value is not highlighted or placed in a position of screen importance.

- The two variables that appear outside of the matrix (and appear important) are the number of students/year and the number of tasks NOT trained to standard. These values can not be manipulated within the module and are informative only. (OSBATS must be exited to change this number of students trained in one year constraint value.)

- Values that hold positions of importance in the matrix display are Cost to Train and Hours of Training Required. These are based on the number of students trained. No method of changing the number of students to be trained is possible within the running of OSBATS.

- What is done about tasks that are not trained on any devices? Since the example data does not contain tasks that cannot be trained, no explanation of how to view these tasks is provided. Will the tasks be presented in the Results By Task option, or since they have no assigned training device, will they not be displayed? The documentation does not address this area.

- The TDSM operates with the full task cluster (default). The reasons why the default task cluster is needed in this module are not explained.

- The complex definition and presentation of Reiteration function is too confusing to be used well.

- The module's levels are not transparent. The bottom seven lines, which are intended to portray system options, are continually changing. It is difficult to determine if all module screens have been viewed.

- There are several problems in the presentation of the graphs. First, the x-axis continually changes its scale. The x-axis is a time scale and the end point changes with each task, making it difficult to visually compare graphs.

- The second problem is in the presentation of the entry level performance. Entry level performance is displayed as a single broken line that travels across the entire time scale. If training is taking place and time is a dynamic variable then entry performance does not travel across the whole training time dimension. Entry performance would be single point not a line.

- The third problem concerns the graphical representation of a training task that requires more than one training device. The documentation does not explain why a second number appears in the graph (numbers represent training devices). The location of the second digit must have some value.

- The present method of presentation does not make use of the inherent advantages to displaying information graphically.

Resource Allocation Module The Resource Allocation Module (RESAM) provides results similar to those of the TDSM. However, the methods differ in two respects: (1) the TDSM assumes cost as a linear function of training device usage, while the RESAM breaks cost down into a number of linear segments, (2) the RESAM allows the user to specify constraints on training device use. Constraints may specify the maximum time that a device may be used or the minimum performance level for which a training device may be employed on a specific task. The RESAM involves great computational complexity and calculation time may be ten times longer than in the TDSM. This increased computational load limits the amount of user interaction possible.

Before operation of this module, four inputs must already exist: (a) a created FMS task cluster from the SCM, (b) a saved ifout package from the IFSM, (c) saved candidate training devices as designed in the FOM, and (d) a saved training system package from the TDSM.

This list outlines the specific inconsistencies and difficulties a user will encounter when operating the RESAM. A few of the comments and questions are answered in OSBATS documentation.

- Prior to this module, entry into a module resulted in the listing of the module's main menu. The Problem Selection Menu appears first before the main menu.

- Once a problem package has been selected, the program should return to the module's main menu; this does not always happen. In one out of three occasions the program does not return to the main menu, but stays in the program selection menu after confirmation of selection. A mouse click on a selection labeled Return is required. There is no consistent method adopted by the module.

- The cost value in the overall results display is not highlighted or placed in a position of screen importance. The two variables that appear outside of the matrix (and appear important) are the number of students/year and the number of tasks NOT trained to standard. These values can not be manipulated within the module and are informative only.

- The graph display format is identical to one found in the TDSM and contains the same difficulties.

- The module has a presentation element that may be difficult for users to understand. When a training system package is optimized, sometimes the problem is too large and the model requires additional memory. The program informs a user that computations must be done off-line. Two screens instruct users to exit the module and complete computations off-line. This is an unexpected difficulty in the programs operation. The process of computation and reentering OSBATS RESAM can take over 15 minutes. This is a long down time for a system user. The user may not feel comfortable with exiting the program.

- The last option to be exercised in the RESAM is the Save Constraints option. The user's manual does not document how or why this option should be used. When it is selected, the display prompts the user for a name. Where this will appear is not specified. Although through trial and error it was determined that it appears in the package selection menu as a new option.

Developing OSBATS Under a New Domain

This section describes an attempt to apply the OSBATS model to a new domain, tank gunnery. The domain of tank gunnery was chosen because of IST's involvement with the Army in this area, and because IST has several devices designed to train tank gunner skills.

The SIMNET device is a full mission trainer that simulates all crew positions within a tank, gunner, loader, driver and tank commander. SIMNET is designed to train tank crew interaction, thus the fidelity of the gunners position is less then a full physical representation. No performance measures are immediately available to the trainee.

The Videodisc Gunnery Simulator (VIGS) is a part-task, table top gunnery trainer that can be configured to train both M1 and M1A gunnery operators. It is a medium fidelity device equipped with multiple sighting options which can simulate the gunners primary sight (GPS), the gunners auxiliary sight (GAS) and the thermal imaging system (TIS). VIGS operates using present off-the-shelf technology in a minicomputer environment. Target scenes are presented by computer generated imagery (CGI) graphics located on videodisc equipment. VIGS adequately simulates the other crewmen located in the actual equipment. Composite measures of performance are presented to the trainee by a embedded CRT located on the device. Overall composite scores are available after each engagement.

The TopGun prototype is a single player, arcade style trainer for M1 gunners. It is a low-cost, part-task trainer which employs computer generated graphics (CGI) and sound effects in the training scenarios. The reticile area is an 11-inch diameter circle which depicts the battle field as viewed through the gunner's sights. The device provides immediate feedback and performance measures on screen. The device adequately simulates all missing crew positions of an operational tank except the driver. Most variables within the threat environment can be changed through simple menu driven control screens, via the Field Modification Initialization Parameters (FMIP).

The VIGS and TopGun devices were used as examples of existing training devices. Cost data and fidelity level information were obtained for both devices and then encoded into OSBATS model data formats.

The first problem encountered with changing domain is the lack of documented procedures. No method has been developed to help a user collect data in a new domain area. Thus, methods and procedures had to be developed iteratively. Without the help of the original model developers this effort would not have been possible. Before application of OSBATS in the gunnery domain, relevant data had to be collected on the following data items:

1. Task Data
2. Device Data
   a. Main Device data
   b. Device IF link data
   c. Device FD link data
3. Student Data
4. Instructional Feature Data
5. Fidelity Dimension Data

The task data was collected from Morrison and Hoffman (1987) and modified to create a critical task list for IST's purpose. This supplied IST with a good cross section of tasks for system coding. The tasks included procedural operations, emergency equipment operations and standard target engagement tasks.

The data required by OSBATS is available only through extensive investment of resources and time. Research must first be conducted which specifies relationships between tasks taught on the training device and the fidelity required to train these tasks effectively. Presently, data must be

obtained through interviews with subject matter experts (SMEs), historical contract information, and direct observation. The data collection effort was not designed to obtain precise information. Approximate values were used when data was difficult to obtain. The focus of applying OSBATS to a new domain area was not to obtain accurate results but to document the difficulties encountered with data input. All attempts were made to gather accurate data, but extensive data searches were cut short due to time constraints.

Cost figures relating to VIGS and TopGun were obtained through structured interviews with SMEs at IST who were familiar with the two candidate devices. These interviews were aided by the use of the training device cost survey. Attempts to obtain cost data from the manufacturing source failed. Manufactures were reluctant to distribute cost figures. Thus, training device cost surveys were completed by two SMEs. The results were averaged and used as system inputs. The Device IF and Fidelity Dimension link data were obtained through the training device capability survey. Attempts made to obtain the student data from the armor school located at Ft. Knox failed. Instead, information from an interview with an army SME estimated student data. The data obtained was recorded and structured for input into the model's data base. Appendix B lists the flat file formats in which the data was ultimately encoded. The lists include the file names and directories where the flat files can be found. Short descriptions are provided of file content.

Integration of an off-the-shelf data base system FOCUS was used as an interface between the model and resident and non-resident data input. This system is not considered part of OSBATS. The DBMS is thus treated as an extra and was not modified to meet the specific requirements of OSBATS. Since it is used only as an outside source (separate environment), interface problems arouse. Quick fixes of the data base system were done iteratively throughout the project. The use of FOCUS was necessary during development.

This list contains the abbreviations found in FOCUS followed by the versions contained in the model's data creation appendix. The questions indicate the information required by the data cells.

RQABS: (ABS RQRD) Does there exist a absolute requirement for simulating this task? Does there exist a reason or need that this task must be trained using simulation. Input value of either a 1 for yes or a 0 for no.

RQSPEC1: (SPEC WX) Does this task require a special set of weather effects to train it accurately? Input values of either 1 for yes or 0 for no should be used.

RQSPEC2: (SPEC SIT) Is there a set of special situations that must be met in order to train this task? Input values of 0 or 1.

RQSPEC3: (SPEC EQUIP) Is there any special extra equipment needed to perform this task? Input values of 0 or 1.

RQTNG1: (TRNG EFFECTS) Are there any special training effects needed? Input 0 or 1.

RQTNG2: (TIME SAVE) Is there a time savings associated with using simulation? Input 0 or 1.

TENTRY: (ENTRY) This is the task learning point, how do we expect the student (0.0-1.0) to perform the task at the beginning without any training.

TSTD: (STD) This is the criterion performance we expect to reach after training has taken place. Input 0.0 - 1.0 value.

CLASS: (CLASSRM HOURS) This is the amount of classroom training that would be required if no simulation, actual equipment or TD were use in the training system. This input value is an open ended figure and can range from zero to whatever.

NOFLT: (NON-FLIGHT) This is the amount of non operational time it would take to train a task. This is also an open value.

FLT: (FLT) This is the number of hours that it would take to rain the task on the actual equipment. This is also an open value.

SETUP: (SETUP) This is an estimate of the number of hours it takes to prepare and get to a point where the actual equipment (AE) can be used for training. Open value.

OTHERS: (OTHER EQUIP) Are there any other equipment costs associated with running of the AE? Open value.

Lack of definitions of data cells was not the only difficulty in the first stage of data input. In two cases no reference point was given to required input values. To answer the TSTD and TENTRY field entries a scale from 0-1 was created. This was the first of many assumptions that needed to be made in order to establish OSBATS in a new domain. It must be stated that the values and labels given in this field were based on general principals, no objective method of placing tasks along this continuum was used. The second assumption made about the answer of the input field RQSPEC3 (Special Equipment needed). It was decided that targets were to be considered as special equipment since firing of a tank requires a target either moving or non moving.

The next information expected by the FOCUS DBMS was information about specific devices. The first set of data asked for is information about Main Device Data and asks for field inputs for five different areas:

1. INVEST: Investment Cost= The dollar amount spent at a specific point in time for a simulator from initial conception through delivery, including initial training and facilities preparation.

2. FIXED YEAR: Annual Fixed Operating Cost= The annual fixed costs of maintaining a simulator are those which occur even if no student training is conducted.

3. VAR. HOUR: Variable Cost Per Hour= The Variable costs represents those expenses for maintaining a simulator that change as a function of student utilization.

4. L.C.: Life Cycle in Years= The projected useful life in years of the training device as estimated at the time of procurement.

5. UTL: Maximum Annual Utilization= The maximum number of hours the training device is utilized in one year.

The main device information pertains to existing devices that may be used to train the tasks that the training device being designed will train. This information is used in the TDSM to help in determining ideal system configurations for training. This information will allow the model to decide if present devices will out perform or cost less then developing a new system. FOCUS next asks for more information about these devices. Information about instructional features and the level of fidelity of the device. Each fidelity dimension must be addresses and a benefit value imputed into the field. The benefit values for different levels are found in the <u>Model Data Collection and Utilization Guide</u> (Willis, 1988). The dimension level is determined with the help of the training device capability survey. Then the level is matched with benefit value in a matrix. The benefit value is then imputed into the data field.

The next two pieces of information required by the DBMS are (1) Device-Fidelity Dimension link data and (2) Device-Instructional Feature link data. These two categories of data help describe the existing training devices's features to the model. The only difficulty with this portion is that

there exists no information documented on what the input fields are. The appendix on FOCUS shows only the screen that calls the IF or FD. No mention of what type of information needs to be placed in these files. The instructions tell the user to enter the IF or the FD name and correct domain name before input is allowed. There are no instructions as to which IF of FD might be appropriate. Through working with the model developers, it was found that all features that were present on the existing TD needed to have a value of one placed in the input file. This information is used in the model to help assign IF and FD of the TD to tasks in the data set.

The model defines tasks as having qualities that require different IF and FD to train adequately. By defining the existing TD in the same terms, comparative judgments can be made between all TD's. This is done primarily in the Training Device Selection Module and in the Resource Allocation model. Since all IF and FD need to be addressed for each device, there are a total of 33 different entries required. All 12 fidelity dimensions must have a value place in the input field. This value is based on a matrix of the 12 dimensions and a total of 48 levels within the dimensions. The values of each level can be found in the model data collection and utilization guide, Appendix A. If the TD does not contain the FD, a value of 0 must be used. If no FD value is entered a value of zero is used as a default. This is not stated anywhere. The 21 IFs must each be addressed and a value of 0 (the device does not contain the IF) or 1 (device does contain the IF) must be placed in the input field.

The next form of data is Student data. This asks for information regarding student costs. The input file ask for values for 9 different fields described by acronyms which were not defined (e.g., "LC_MUL").

Instructional Features (IF) can be modified through FOCUS. The IF is addressed using its acronym, a new benefit weight and cost value can be imputed. New IF can be added within the FOCUS shell, but to have the expert system address the new IF requires changing the rule base. The information about an IFs value is only the benefit weight and cost of the feature. This appears on the surface to be too simple when compared with the other data types. Assuming that the values associated with IF are based on a wide range of systems, making these values generic across domains would not reflect accurate development cost and benefit values.

The FOCUS interface system allows manipulation of the FD found in the model. The DBMS allows us to view each acceptable FD and to change its values. There are 11 different input fields labeled by acronyms not defined (e.g., "TECH1").

Altogether, 60 pieces of data are required for each task, and neither the Use's Guide nor on-line documentation is adequate for identifying the appropriate data for entry. A good user's data collection manual could be devised quickly and easily. To make obtaining data for a new domain easier, each of the data input fields in the FOCUS interface could have a section devoted to explaining how to collect specific field inputs. The importance of the data field being entered, and where and how the model uses this information could be provided.

The only problem lies in applying the data gathered in a new domain to the OSBATS model. Presently, there exists no formal method of having OSBATS recognize data from any domain then rotary wing aircraft. New expert system relational rules would need to be developed for new domains. No procedures exist of how to change the rule base, nor are estimates as to the amount of alterations that would need to be made available.

Operation of OSBATS in New Domain

After all data input steps had been completed, and file formats were compatible, running of the OSBATS model was attempted. It was hoped that after installation of data items that the model would function. Initially the system loaded the data base with no difficulties. The first module was initiated with no system failures. The model continued to work until the Training Device Selection module was reached. At this point a "floating point" error was reached. This error is produced by division by zero, or a zero value was placed in a data field position where a zero is an unacceptable value.

During operation of the Fidelity Optimization Module, the system began delivering inconsistent responses in benefit weight allocation. Certain fidelity dimension were given benefit weights that did not make sense. For example, the fidelity dimension seat motion,"seat shaker" was rendered a high benefit value, and the Visual Resolution fidelity dimension was assigned low benefit values. It was determined that this discrepancy was caused by a difference in data base file organization between that model and the database management system used to create the data base. The expert system which presides over OSBATS expects file formats of fidelity dimension to match its own templates. This template place fidelity dimensions in an arbitrary order. While EXSYS ordered data inputs in alphabetically. The data base management system constructs overall file formats that conform to model expectations, but places input values in incorrect matrix tables.

Another model convention made itself known through this interaction. The task benefit weights are based on time savings that occur when this task is trained using a simulator, rather then the actual equipment. This weight is calculated using information directly from task descriptions and a baseline full mission simulator (FMS). The FMS simulator is compared to the actual equipment. The model assumes that the baseline full mission simulator is the fifth training device that the user inputs into the data base, since all baseline data is taken from the training device in the fifth input position. This device must be the best available full mission simulator. The documentation does not explains that the fifth device imputed is the key reference point where all model benefit weights are calculated. If the information is incorrect or a part mission device is described as the fifth device, then benefit assignment will be incorrect. The model will then produce erroneous results which the user may not notice.

Input sequence also affects the number eight position in the device list. The model assumes that the eighth position is the fidelity definition of the actual equipment. Any other training device imputed in the eighth slot will be treated as the actual equipment. Benefit values will be computed using the eighth input position as the values for the models baseline comparisons. Erroneous results will be produced if the eighth position is not actual equipment information.

The encoding of data for the application of OSBATS to the gunnery domain was incorrect. Accurate data was collected and imputed into the Data Base Management System (DBMS) correctly. It was the DBMS organizing of the data fields in alphabetical order that caused the discrepancy between model templates and DMBS organization patterns. Time constrains limited correction of the file formats. Quick fixes were devised, but reinstallation of all model data was preferred. The reordering of file formats by hand presented to many opportunities for error; any mistakes made would be difficult to locate later. Thus, the lack of coordination between the model and its DBMS has caused difficulties in developing an adequate data base.

The objective of applying OSBATS to a new domain was not to establish accurate results for the design of a training system. Instead, application of the model was intended to illustrate any difficulties or problems that would be faced during creation of new domain data. The simulated application of OSBATS to the gunnery domain has provided illumination of the problems encountered. More realistic expectations of systems capabilities were established.

A new domain application is not a simple task and can not be accomplished without the aid of the model developers. No procedures or documentation exists to aid a user in the development of new domain information. This lack of documented procedures limits the usability of OSBATS. Although the last two modules were not successful in operation, the new domain development effort proved the concept to be viable.

ASTAR (AUTOMATED SIMULATOR TEST AND ASSESSMENT ROUTINE)

The Automated Simulator Test and Assessment Routine (ASTAR formerly DEFT) is an automated decision aid designed to assist an analyst in evaluating the effectiveness of a training device or training system. ASTAR uses generally accepted training principles, performance feedback, and similarity of the training device to operational equipment to evaluate the effectiveness of the training device or system. ASTAR converts information and judgments provided into a forecast of the effectiveness of the device or system.

ASTAR has three levels of evaluation which are dependent on the level of detail (data) available. The decision about which level to use is dependent upon the amount of information available. Data, in varying degrees, must be collected regarding the training device, training method, operational equipment, required performance levels, tasks to be trained, and information about the trainees themselves. ASTAR computes several effectiveness scores for comparisons between devices or systems. An Acquisition Effectiveness score and a Transfer Effectiveness score provide a basis for comparisons of what is learned on the device and what remains to be learned on the job. These scores are combined to produce a summary score of Training Effectiveness.

Four major analyses are conducted during an ASTAR training system evaluation: Training Problem, Acquisition Efficiency, Transfer Problem, and Transfer Efficiency.

The first two deal with the trainee Acquisition Effectiveness of the trained task. The Training Problem defines the skill and knowledge deficits that the trainees posses before training begins. Ratings of how difficult it will be for the trainees to overcome these identified skill and knowledge deficiencies are established. The next analyses examines Acquisition Efficiency. The quality of training provided by the device is studied by analyzing the instructional features and training principles that have been incorporated in the device to help trainees overcome their deficits.

The third and fourth analyses address the Transfer Effectiveness of the learned tasks to the actual equipment. The Transfer Problem determines the deficiencies trainees will have with respect to operational criterion performance after they have achieved criterion performance on the training device. The difficulties in overcoming these residual deficits is determined and any effects of physical and functional dissimilarities between the training device and the actual equipment are assessed. The fourth analyses is Transfer Efficiency; it indicates how use of the training device will promote transfer of the learning task that has occurred to the actual equipment.

Hardware

The ASTAR software is contained on one 360K floppy disk. The data must be saved on a second disk. The program is written to run on an IBM (or compatible) equipped with dual disk drives or a hard drive and one disk drive.

Documentation

ASTAR has a users manual delivered with the software. The ASTAR users manual (AIR 1988) contains detailed instructions on how to develop needed task lists, collect information for database construction and the construction of the data base. Instructions are clear on how to conduct the analysis and how to interpret the rating results. An example of an existing data base is included in model software to help illustrate model use. No further documentation is needed to adequately implement the model. The instructions are written with simple direct sentence structure for easy comprehension.

Data Requirements

ASTAR data requirements include descriptions of actual (parent) equipment (AE), the training device (TD), the operational tasks, the training task, the trainees, and the utilization of the trainer. The data requirements are flexible and depend on the level of analysis being conducted. For an ASTAR level 1 analysis, general descriptions of the above categories are all that is needed. An ASTAR level 2 analysis requires detailed descriptions of the training device, actual equipment, and task lists for TD and AE. Data on what types of skills and knowledge that are required to perform the tasks, student characteristics and device utilization plan is also needed. An ASTAR level 3 analysis requires a deeper level of detail. In addition to all the above information, specific descriptions of the locations and functions must be provided for controls and displays. The tasks using these controls and displays must be identified. Also, specific information on training performance criteria, detailed demographics of trainee population, and knowledge of planned device utilization pattern is needed.

ASTAR's most complex level of analyses (level 3) requires data on specific tasks, subtasks, displays, and controls to be encode into the data base using numerical labeling. These data items must be encoded into the data base in strict format styles. The other data categories can be in any format, assuming model users understand the data and can make rating judgments based on the data. The omission of any one or two data items does not stop the system from operation, but reduces the level of analysis possible. This flexibility allows the system to be used at any level of data detail. Creation of the data base uses information that is readily available: detailed task lists, and functional descriptions of the actual equipment. The other data items in the data base are specific to the training device being designed, as the level of design sophistication increases, the amount of relevant model data increases as does the level of ASTAR analyses.

Approach

ASTAR's approach is simplistic; it breaks down the training problem into easily understood intervals: the Training Problem, Acquisition Efficiency, Transfer Problem, and Transfer Efficiency. The model becomes a personnel management tool when used in the method described in the users manual. Ideally, ASTAR analyses should be conducted by several raters. The user's guide suggests a mix of individuals from differing fields who are familiar with the training design process. This mix should include three to five persons from these disciplines: Instructional Technology, Psychology, Engineering and Human Factors. The diverse backgrounds will provide different and valuable perspectives. The analyses procedure becomes an exercise in compromise.

The information needed for any given level is collected, documented and delivered to all design team members. Each reviews the information individually, then all members meet to discuss the major assumption that need to be made regarding the eight major analyses. The raters conduct their analyses independently, recording their reasoning behind chosen ratings. The team meets to discuss and compare results and to determine the reasons behind any difference in judgments. The raters re-assess their judgments, striving for consensus. This iterative meeting process continues until all members find an acceptable design.

This method of bringing people together is the strength of the ASTAR model. This ensures that design suggestions from Human Factors experts, IT, Psychologists and Design Engineers are considered throughout the design process. The model involves each member of the design team in the design process, and helps to clarify assumptions made by different disciplines. Often design team members have difficulty communicating the importance contributions available in their individual disciplines. Thus, ASTAR facilities necessary communication between the diverse disciplines responsible for training device design. On-screen definitions for each question are provided to help ensure that people are interpreting the questions on the same level.

Validation Efforts

ASTAR has been validated in multiple areas using different raters and has demonstrated good interrater reliability and acceptable face validity. The implementation of the system, including data collection can be accomplished in a reasonable amount of time. ASTAR can be used to compare devices in early concept formulation phases, and can be used to evaluate the effectiveness between those recently developed and those in latter stages of the design process (Rose, Martin, and Wheaton 1988).

CONCLUSION

ASTAR is presently being fielded and has been validated in different areas of training device design. ASTAR and OSBATS can not be compared directly. OSBATS is a prototype. It is too early in the development of OSBATS to determine its final weaknesses and strengths as compared to other models. OSBATS is potentially a powerful design aid if the shortcomings relating to strict data requirements and lack of documentation can be overcome with further development.

REFERENCES

American Institutes for Research (AIR) (1988) ASTAR User's Manual. Washington, D.C.: Author

Feigenbaum, E., Mc Corduck, P. & Nii, H.P. (1989). The Rise of the Expert Company. New York, NY: Times Books.

Foley J. D. & VanDam, A. (1982). Fundamentals of Interactive Computer Graphics. Reading, MA: Addison-Wesley.

Gilligan, E. L., Elder B. L., & Sticha, P. J. (October 1988). Optimization of Simulation-Based Training Systems: User's Guide. Research Produce 88-30. Alexandria, VA: Human Resources Research Organization.

Hays, R.T. & Singer, M.J. (1989). Simulation Fidelity in Training System Design: Bridging the Gap between Reality and Training. NY: Springer-Verlag New York Incorporated.

Hofer, R., Ozkaptan, H., and Kincaid, J. P. (1987) Training Systems R&D. Proceedings of the 9th Interservice/Industry Training Systems Conference, Washington, D. C., 32-41.

Morrison, J.E. & Hoffman, R.G. (1987) Requirements for a Device-Based Training and Testing Program for M1 Gunnery: Volume 2: Detailed Analyses and Results. Alexandria, VA: Human Resources Research Organization.

Ragusa, J. M., Barron, A. E., Vine, M. & Schieffelin, T. (May 1989). Interfacing and Using Small AI Expert Systems with Interactive Laser-Optical Technology. Report TBD. Orlando, FL: University of Central Florida - Institute for Simulation and Training.

Rose, A.M., Martin, M.F. & Wheaton, G.R. (1988) Forecasting Training System Effectiveness: Review and Analysis of ASTAR. Washington, D.C.: American Institute for Research.

Sherald, M. (July/August 1989). "Neural Nets Vs. Expert Systems." PC AI, 10-15.

Shneiderman, B. (1987). Designing the User Interface: Strategies for Effective Human-Computer Interaction. Reading, MA: Addison-Wesley.

Silverman, B. G., ed. (1987). Expert Systems for Business. Reading, MA: Addison-Weseley.

Simon, H. A. (1957). Administrative Behavior. New York, NY: Macmillan.

Singer, M. J., and Sticha, P. J. (1987) Designing Training Devices: The Optimization of Simulation-Based Training Systems. Proceedings of the 9th Interservice/Industry Training Systems Conference, Washington, D.C., 26-31.

Singer, M. J. (July 13, 1989). Personal interview.

Smith, S. & Mosier, J. H. (1984). Design Guidelines for the User Interface for Computer-Based Information Systems. Bedford, MA: MITRE Corp.

Sticha, P. J., Blacksten, P. J., Ric, H., Buede, D. M., Singer, M., Gilligan, E. L., Mumaw, R. J. & Morrison, J. E. (October 1988). Optimization of Simulation-Based Training Systems: Model Description, Implementation, and Evaluation. Final Report 88-26. Alexandria, VA: Human Resources Research Organization.

Stock J. A., Gilligan, E. L., & Sticha, P. J. (October 1988). Optimization of Simulation-Based Training Systems: Programmer's Guide. Volumes 1 & 2. Research Product 88-29. Alexandria, VA: Human Resources Research Organization.

Waterman, D. A., (1986). A Guide to Expert Systems. Reading, MA: Addison-Wesley.

Willis R. Optimization of Simulation-Based Training Systems: Model Data Collection and Utilization. (June 1988). Winter Park, FL: Eagle Technologies, Inc.

APPENDIX A

USER'S SESSION ON OSBATS

This appendix contains a detailed description of a user's session on OSBATS. Each of the five modules are discussed in length.

APPENDIX A

USER'S SESSION ON OSBATS

INTRODUCTION

After the rule base is loaded, the Module Selection Menu is presented. Any of the five modules can now be examined. Modules may be accessed in any order, depending on the needs of the user. However, the Resource Allocation Module (RAM) is only meaningful after a training device package has been developed from the Training Device Selection Module (TDSM). For the purpose of this report, all modules will be reviewed in sequence of main menu ordering, since this is the logical progression of modules during the designing of a training device. The modules follow this order:

- Simulation Configuration Module (SCM)
- Instructional Feature Selection Module (IFSM)
- Fidelity Optimization Module (FOM)
- Training Device Selection Module (TDSM)
- Resource Allocation Module (RESAM).

The Simulation Configuration Module addresses task selection, the Instructional Feature Selection and Fidelity Optimization Modules address sophistication and cost trade-offs, and the Training Device Selection and Resource Allocation Modules addresses the allocation of training resources.

SIMULATION CONFIGURATION MODULE

The Simulation Configuration Module (SCM) classifies all training tasks into three clusters:

- tasks to be trained on a full mission simulator (FMS)
- tasks to be trained on a part mission simulator (PMS)
- tasks to be trained on the actual equipment (AE).

The SCM examines each task to determine the need for simulation and the cost-effectiveness of using a simulator to train rather then the actual equipment. The segregation of tasks is based upon task training requirements and the potential operating cost savings obtained with simulation training.

Once the SCM is selected from the main module selection menu, three options are available:

- Display Results
- Save Results
- Module Selection Menu

The first option within the module is Display Results. This option contains the following options:

- Graphical Summary
- Individual Tasks
  - Descriptions
- Tabular Summary

The first screen encountered in the Display Results option is the Graphical Summary. This graph illustrates the partitioning of the task set into task clusters. This partitioning is based on the cost-effectiveness of training on a general class of simulators compared to training on the actual equipment. As Chart 1 shows, there are four different variables that can be manipulated within this display:

- Training Requirements index = 0 - 1 (TRI)
- Operating Cost Savings = 0 - 1 (OCS)
- Cost Boundary Slope = values -40,-20,base,20,40.
- Task Complexity Line = 0.1 to 0.9

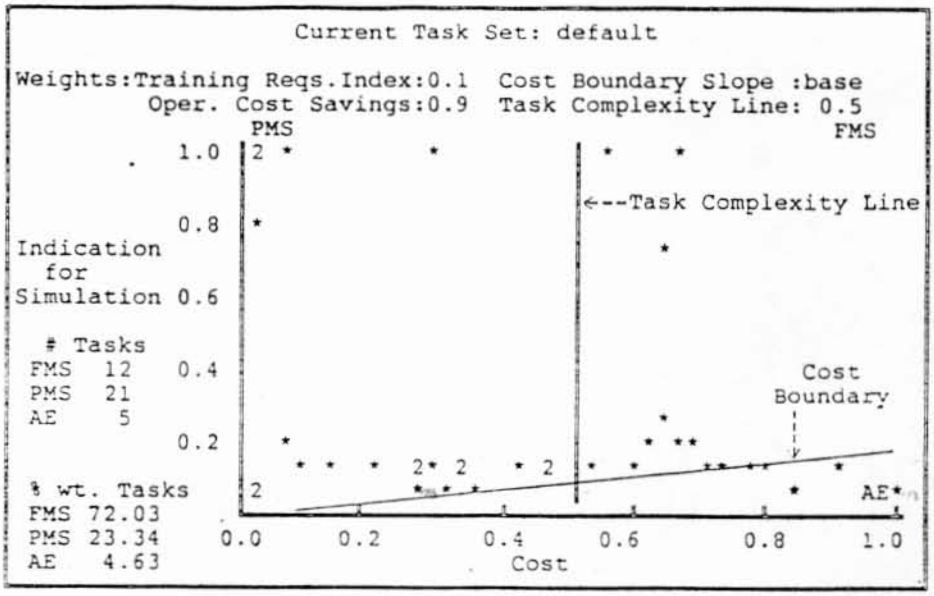

Chart 1 Graphical Summary (SCM)

The training requirements index and operating cost savings are dependent upon each other. The combined value of these two variables must be equal to one. As the value of one increases, the value of the other decreases. Users are instructed to adjust the weights to place greater importance on one of these indices. The documentation does not thoroughly describe what these two manipulations are intended to accomplish, nor are accurate definitions provided. In an example provided in the guide, the module has already allocated a cost savings value to each task (this is needed to obtain the indication for simulation value). How is this done so early in the analysis? Why does a dependency between the two variables exist? Neither of these questions are suitably answered by the documentation.

The cost boundary slope corresponds directly to the near horizontal line on the graph which delineates tasks that fall into the actual equipment task cluster category. The cost boundary slope is determined by equating the total operating cost savings over the equipment life-cycle (which is obtained using a baseline training device with the development cost). The values range from 40 percent below to 40 percent above the baseline (nominal value).No definition of baseline is provided. (The values increase and decrease in steps of 20.) The available documentation does not adequately explain the reason for the baseline or how it is derived.

The task complexity line corresponds directly to the vertical line in the center of the Graphic Summary. This line separates the PMS from the FMS task cluster category. The placement of this weight

is determined by the user. For example, if a task lies on the PMS side and the user feels that a PMS would not give all the cues necessary for training this task, the user would adjust the weight values until the task is on the FMS side of the line. Only the line moves, no changes occur in the task location or the weights.

The documentation does not adequately explain why any of these four weight values should be changed. Nor does it explain what the implications, effects or consequences on further models will be if changes are made.

The Graphical Summary displays information on the number of tasks that fall into each category, the percentage of task weights that fall into each category, and provides a method of examining individual task values for cost and indication for simulation. The display allows access to each point on the graph.

Selection of a point changes the task for which information is provided. The x-axis of the graph displays development cost of the simulator and the y-axis displays the indication for simulation. The development cost is determined from the task cue and response requirements; the indication for simulation is calculated from the requirements for training the task and the potential operating cost savings associated with simulator training. These values are the key to the clustering of the tasks, yet no reference is provided explaining where the cue and response requirements or the potential operating cost savings figures can be found. (The basis for tasks values is found in the individual task data.)

Model conventions state that the bottom seven lines are intended for display options, yet the graphical summary display uses the bottom left hand corner of the screen to display task information. The amount of information packed into the first display a user will view results in immediate impression that the system is complex and user unfriendly. The amount of information presented in the display is useful to an expert system user, but a new user attempting to gain model understanding might not be able to assimilate the large array of information.

The next screen in the Display Results option is the tabular summary display. The tabular summary display lists the tasks assigned to each of the three cluster categories, AE, FMS, and PMS. These are complete lists of all tasks that fall into each category. An option within this display isolates all tasks in a category and shows their normalized cost and indication for simulation value. The individual task values can be examined in more detail by selecting the Individual Task Data option. The individual task data tables show the intermediate results that underlie the overall module results. The origin of the fidelity dimension requirement values, which cost to train is derived from, are not revealed.

The final screen to view is the Sensitivity Analysis. This screen allows the user to examine the effect of changing the cost-boundary slope. The display presents two graphs with the y-axis unlabeled. Even after a careful reading of the documentation, it is difficult to determine what information the user is expected to draw from this display.

After all screens have been viewed and the user has determined that the task set has been divided into categories correctly, the saving of the task clusters ends this module. From within the Display Results screens, the user must exit back to the main module menu and select the Save Results option. The screen presents a summary screen of the values manipulated in the module. Saving is accomplished by placing the curser over a label titled Save Cluster. The system does not ask for a label, but instead will place its own numbering scheme on the clusters. The system does not allow the user to chose the label for the cluster of tasks.

The output values produced in this module will be input values for the Instructional Features Selection Module and the Fidelity Optimization modules.

INSTRUCTIONAL FEATURE SELECTION MODULE

The Instructional Feature Selection Module (IFSM) begins the training device design process. Instructional features are elements of training devices that can improve training efficiency on individual tasks. This module requires a task cluster obtained through the SCM or defined by the designer. The IFSM uses a set of rules to compute benefit values for each instructional features on a task by task basis. OSBATS presently contains 21 candidate instructional features. The development costs are matched with individual instructional features to obtain a cost-benefit values. OSBATS then orders the instructional features in descending benefit to development cost ratio. The IFSM then requires the user to select a group of instructional features for further examination in Fidelity Optimization Module (FOM).

Once the IFSM has been selected from the main module menu, five options are available:

- Examine/Edit Task Set
- Examine/Edit Instructional Feature Set
- Display Results
- Save Changes
- Module Selection Menu

The IFSM begins by selecting or creating the task cluster. Selection is done through the Task Cluster Library described earlier. After selection and return to the model main menu, the user should select the Instructional Features option. If no appropriate task clusters exist, the user can enter directly into the IFSM.

The first option within this module is the Examine/Edit Task Set (EETS). If a task cluster has been chosen, the user examines the two categories of tasks (included and excluded) to verify that tasks are grouped correctly. If the user must develop a unique task cluster, independent manipulations of the default task cluster are needed. When in the Task Cluster Library, the user selects the default task list so that all tasks are available for editing. Editing is performed by placing the mouse curser directly over the task that to be excluded and clicking the mouse to move that task. Task clusters must be saved after editing. Discussion of how to save this new task cluster will wait until the next module function is examined.

The third second menu option in the IFSM is Examine/Edit Feature Set (EEFS). This option allows the selection of the instructional features for inclusion in the analysis. The two column format from the EETS is used, with instructional features to be included on the left and those to be excluded on the right. Exiting this option to the IFSM menu places the altered instructional feature list into a working memory.

The fourth menu option in the IFSM is Save Changes. This option allows changes in the task cluster or instructional feature set to be saved. This option is exercised only if the task cluster or the instructional feature set is altered. The program will not allow the user to exit the IFSM if changes have been made but not saved. Although this monitoring procedure is helpful, presentation of a Save Changes option in the main menu implies that the operation is required.

During the saving procedure, the program requests a user defined name, less then eight characters, and a description of the cluster set. The saved task cluster will appear in the Task Cluster Library. When saving changes to both IF and task clusters, only one name and saving procedure is required.

The instructions are not clear as to what happens to the instructional feature set changes. The changed grouping does not appear as a selection choice, but is embedded in the task cluster.

The saving of altered packages can wait until all manipulations have been completed. The reasons for not immediately saving the newly developed lists have not been supplied, nor are adequate directions given as to why the Examine/Edit options might be implemented.

The formats of the Examine Edit option clearly present where the tasks will fall. If task clusters previously defined with the SCM are changed at this point, assumptions made in the SCM are invalidated. This is not explained in the documentation. A set of user cautions statements is needed to inform the user of the full impact of the editing choices he is making. Changing task clusters early in the model violates only a few working assumptions; however, as deeper levels of the model are reached, changing a task set will result in greater consequences. These consequences will be explained in this report when they occur in future modules.

The third menu option in the IFSM is Display Results. The IFSM will use the chosen or newly developed task cluster to calculate which instructional features offer the most benefit per cost for training the specified tasks. Within this option are three major selections:

- Summary Display
- Graph Display
    - Benefit/Cost Graph
    - Benefit Graph
    - Cost Graph
    - Cumulative B/C Graph
- Assignment List

The first screen encountered in the Display Results option is the summary display. Within this display, task weights and feature weights can be excluded or included in the calculations. Instructions explaining the consequences of including or the excluding of weights are not provided. When task weights are included in the calculations of total benefit, task weights are handled as decimals, and when excluded, task weights are handled as whole numbers. This discrepancy does not change the calculations since the values are eventually normalized. It only becomes a problem when users attempt to compare the difference between excluding and including task weights.

The screen layout position of task and feature weights is reversed. Changes in task weight inclusion, located on the right side of the screen, result in changes of column values located in the left side of the screen. Inclusion of instructional feature weights takes place on the left side of the screen while column changes appear on the right side of the screen. This screen variation makes it difficult to see what values have changed when weight are excluded. Inconsistencies in labeling also exist. The screen labeled Summary Display is called up by selecting a menu option titled Results Table

The second menu option graph display portrays four different graphs of relevant information.

- The Benefit/Cost Graph is a bar graph of the numbers found in the B/C column of the summary display table. It shows the B/C ratio in order of decreasing value.

- Benefit Graph is a bar graph of the numbers found in the total benefit column of the summary table. This shows the weighted average of the number of tasks appropriate for each IF. This presentation is also in order of decreasing B/C ratio.
- Cost Graph is a bar graph which shows the values from the cost column in the summary table, the IF are in order of decreasing B/C ratio, producing a staggered graph.

- Cumulative B/C graph is a graph where each point representing a list of IF. The IF with the next highest B/C ratio is then added to this list to produce the next point on the graph. This section allows you to save and choose configurations of IF for later model examinations to be based upon.

Options must be chosen that illustrate the level of IF sophistication the user is willing to accept. The chosen options will be incorporated in the Fidelity Optimization Module (FOM) to help obtain the overall design of a training device. A maximum of seven points can be saved.

The module quickly produces each of the graphs. The option to include or exclude task and instructional feature weights is still available. When changes occur, the order of the colors (each instructional feature is represented by a different color bar) stay the same but the IFs change position on the X axis and in the legend. For example, the blue color bar is always first to be graphed regardless of which instructional feature it represents. The colors are repeated three times each when all 21 instructional features are present. There are more IFs then available colors to graph them. The y axis is not labeled on any of the graphs.

The fourth graph titled Cumulative B/C Graph is the most critical of all screens within this module. Within this screen that ifout packages are saved for use in the FOM. The FOM requires a created ifout for module functioning. This information is not stressed strongly enough. This graph is embedded in the fourth level of a submenu option. The first three graphs present no new information, but show pictorial representations of the summary table. The cumulative B/C graph presents new information about optimal instructional feature packages. As points move along the optimal curve, the instructional feature with the next greatest benefit to cost ratio is added to the grouping. By selecting a point on the curve, a boxed insert is provided which gives information about the number and value of instructional features that make up the point on the graph.

During this module one major system convention has been violated. When the Assignments List menu option is chosen, the submenu choices appear in a field designated as presentation areas of graphs and tables, not menu options. All module options are not displayed continually; when an option is chosen, the option label disappears and is replaced by the title option of the previous display. This results in menu options switching positions throughout the analysis. For example, the option Graph Display is suddenly replaced by the option Assignment Lists. This is due to the limited area of the screen that is dedicated to menu selection options.

With a developed instructional feature package, the user can now move on to the next module.

FIDELITY OPTIMIZATION MODULE

The Fidelity Optimization Module (FOM) continues the training device design process begun with the IFSM. The FOM has two performance goals. The first goal of the FOM is to specify the levels of technical sophistication on a set of fidelity dimensions. OSBATS presently contains 11 fidelity dimensions varying in the number of levels within. Each dimension level has an associated benefit and development cost value. The FOM selects dimension levels to maximize the effectiveness of a training device given development cost. The FOM may recommend several different device configurations at different device costs. The second goal of the FOM is to make trade-offs between technical sophistication of presenting cue and response feedback and technical sophistication of instructional support. This requires the use of the prioritized list of instructional features developed in the IFSM as input.

Once the FOM has been selected from the main module menu, six major options are available:

- Include Instructional Feature Results
- Examine/Edit Task Cluster
- Constrain Fidelity Dimensions
- Display Results
- Save Changes
- Module Selection Menu

Before operation of this module, two inputs must already exist: the cluster of tasks created in the SCM and a saved Instructional Feature package. The IFSM must first be used to develop a design package of instructional features. These design packages should have been saved using unique titles.

The first step in the FOM is to chose an instructional feature package to be used in the fidelity dimensions calculation. The option Include Instructional Feature Results, should be examined. This option chooses an IF package previously created in the IFSM, allowing incorporation of the results from the IFSM into the current analysis.

The display format is clear. Two responses are necessary. The first response is to chose an ifout package with the mouse. (Ifout is the nomenclature used in identifying IFSM packages.) The second response is to confirm the choice by using the key board. This is the second time a keyboard response is required. After confirmation, the main module menu automatically appears; in all other occasions the user must chose the return to module menu option provided.

A few inconsistencies or possible problems exist in this first module option. Only one IF package can be examined at a time with a given task cluster. A comparisons between Instructional Feature

packages is not possible. Examining an ifout package after they have been created is not possible. The differences between two developed packages can not be investigated. When a task cluster is chosen that has not had the IFSM operation performed on it, no option to Include Instructional Feature Results is presented in the main menu. Yet, all other main menu options are still functioning. The option could be retained in the main menu, in a format indicating the option is present but that a problem exists. For example, the yellow border of the option could be replaced with a red border; then by choosing the red bordered option, an explanation for its present status could be given.

The selection to delete an ifout package requires a different procedure when returning to the main menu. First the red deletion box must be selected. The user waits for something to occur, but nothing happens. Next the selection to return to the main menu must be made. Instead of returning to the main menu, a decision to confirm the deletion must be made, then the main menu appears. There is little spacing between the displayed listings, causing possible mistakes in cuing. This cuing problem is compounded by the fact that none of the following screen informs the user which ifout package is being used. If the user changes the ifout package, it is unclear if the first package is replaced or if the new package has somehow been incorporated into the first package.

The second step in the FOM is to Examine/Edit Task Cluster (EETC). This is a repeat of the same option available in four of the five system modules. The same procedures and basic problems exist within this module. The examinations of the tasks is useful for verification that the appropriate task cluster has been chosen. The problems arise when the system allows users to interact and change this task list. If task lists are changed, many values will typically change. The benefit values within the IFSM were used to select the IF packages the FOM depends upon. If the task cluster is changed, values change and previous choices made may become invalid. If changes are made in the task set, then changes must be made to the IF packages.

The third step in the FOM is to Constrain Fidelity Dimensions. This selection lists all Fidelity Dimensions (FD) and dimension levels by clicking the mouse on the dimension. Each dimension level also has a benefit and cost value that will be displayed. The user is asked to go through all the dimensions and to exclude those levels of each dimension that are not useful. It is not necessary to save constrained levels at this point; the excluded levels are highlighted in red to indicate exclusion. This elimination of fidelity dimensions and levels is important to obtain a meaningful analysis, yet there is little relevant documentation in the user's guide.

When the dimension levels are presented with their benefit values, a set of on-screen instructions appear. This is the first time, other then when saving a package, a set of instructions is provided. This does not follow prior system formats.

All manuals report that only 11 fidelity dimensions exist, yet when an ifout is chosen, 12 dimension levels appear (inst_features). This new dimension can contain up to five levels. Th impact of this in terms of selection of fidelity dimensions in following module computations is not reported in any of the manuals.

The guide lists actions demonstrating the elimination of G-Force from the model. Instructions are clear of how to click the mouse in the correct place to turn the highlight field to red, indicating elimination. The user is instructed to include the change back to a white field but is not instructed to eliminate it again. However, in later module calculations the G-Force has been eliminated.

The fourth step in the FOM is to Display Results. The FOM calculates which level of fidelity dimension offers the greatest benefit given cost. Within this option are four major menu selections:

- Graphical Display
- Matrix Display
- Package Display
- User Defined Packages

The first screen to be displayed is the Graphical Display, all other options are chosen within the Display Results portion. The Graphical Display shows a benefit to cost curve of optimal training device designs. Each point on the graph represents the addition of the next greatest B/C value when an additional dimension level is added. Through the Increment decrement option in the legend, the user can move up and down this B/C curve. Each point represents a different training device configuration of dimension levels. By selecting a point, the user can view this package in the package display portion. Only those points that fall on the line are considered optimal. Any other design configuration will be less than optimal and will fall below the line.

This display is a good presentation of where each package is on a benefit curve. By moving alone the curve, the user can see the point of diminishing returns. Two submenu options (increment package and decrement package) are used to move up and down the graphs points. (It may have been more consistent to allow optimal packages to be chosen by a mouse click on a point on the graph, since this is the method used in previous modules.)

Within all screen of Display Results, the user has the option to include or exclude both task weights and dimension weights. The weight value of each task is based on the total cost of training the task to criteria using the actual equipment. Tasks that require more training receive a higher weight. When tasks weights are excluded, documentation is unclear as to what happens. The manual states "...if the goal of a training device is to provide familiarization training on a variety of tasks, the task weights... may not be appropriate." How this translates into the analysis is vague. The demonstration of how to change the weights from included to excluded is simple. A mouse click changes the value. When task weights are excluded, more points appear on the optimal device curve. The reason for this increase in optimal devices is unclear.

The Dimension weights are the second manipulatable variables on this screen. Definition of what these are and how they affect module assumptions are not clear. These fidelity dimension weights reflect the relative importance of the fidelity dimensions. An FD has a high weight when tasks with high cure aand reponsse requirements use the dimension in training. Is the assumption being made that for all tasks and training devices that the fidelity dimensions remain in the same order of importance? Does the expert system account for the possible differences of fidelity dimension ordering for different training devices of the same set of tasks? Excluding the dimension weights allows the user to "..compare the number of task requirements that are met for different fidelity dimensions.". This statement conveys little module understanding, or why the inclusion or exclusion would have any effect on module outcomes.

The second module menu option Matrix Display of Packages. There are six different matrix tables designed to show cost and benefit data for all fidelity dimension levels:

- Descriptions - This gives a short one sentence description of each dimension.

- Cost - This gives cost values for each level of all dimensions, it represents development cost only. The costs are estimated from the lowest and highest levels of each dimension and the values in between are interpolated, based on technical performance.

- Benefit values - These are determined by the number of tasks which are appropriate for each level. This comes from the IFSM information.

- B/C ratio - The incremental increases in B/C are shown for levels. Not all levels have values, if the increment from 1 to 2 is greater then the increment from 2 to 3, then level 2 is dropped. The highest B/C are systematically added into the package solution.

- % of cost - this shows, as a % of cost, the highest active level of each dimension. To help determine the relative impact on cost of varying levels.

- % of Benefit - The level of benefit is divided by the highest active level. This information is used with % of cost (above) to understand why certain levels have been chosen.

These descriptions are too short and general to provide a clear understanding of the taxonomy. The end user must have a full and clear understanding of the developers frame of reference. The titles do not convey the same message to all users. The terms fidelity and resolution have been in discussion for years. OSBATS is too specific of a tool to not clearly define its dimension categories.

In specifying cost values, OSBATS is lacking in precision. The value of OSBATS is its cost prediction capability. Empirical data is used to obtain the high and low values, but each point in between is interpolated using technical performance equations. These equations assume low levels of technical performance can be obtained relatively cheaply compared to a level of performance that is closer to the state of the art.

The cost graph gives precise values, but does not mention what is the scale. In other tables the guide mentions that the cost figure is (x 100). The cost figures are static in that they do not change between device. Is there not a dynamic relationship between dimensions and levels of all the features? Do these cost represent this changing relationships between levels of dimensions? For example, if we design a training system with no motion, the visual display mounting and hardware for a given level of fidelity will be different then if a motion base is added. The display, although having the same level of fidelity must now be designed to accommodate motion. This accommodation may increase the cost of the display, possibly to a significant level.

In the third matrix table, benefit values for a given level of any fidelity dimension are given. The values range from 0.00 to 15.70. What is the scale value? What does a value stand for? Higher numbers do indicate level of importance, but why a fidelity level has a value of .09 is not explained in the guide.

Choosing different ifout packages does not change the cost and benefit values for this matrix. These displays are designed to present background data to the user. The matrix values do not change due to design considerations. The only method available to change matrix values are in the exclude and include task and dimension weights. Both the dimension weights and the task weight effect benefit values only, not cost.

The module breaks from standard model conventions stated earlier. When in the matrix table a selection menu is presented which displays the six choices. These choices have black numbers superimposed over a yellow field, consistent with model conventions for available menu choices. The inconsistency appears when choosing a specific matrix table. Once an option is chosen, the fields should switch to yellow numbers on a black field, indicating a option in use. This does not occur. The yellowed option blocks stay yellow even after selection.

The matrix display is the first point where the Print option is usable, yet no explanation is given as to why or how the user would use it.

The third module menu option is the Package Display.
This option gives the dimension level in use by a package. The associated cost and benefit value are also displayed. By selecting the increment or decrement package option in the submenu, different optimal training device design packages are displayed. A cost constraint function displays the nearest package to a imputed cost value, without going beyond the constrained limit. This is the first time a user can impose a cost constraint. The cost is used as a constraint for choosing an appropriate level of fidelity.

The guide introduces a procedure in which certain assumptions are made in cost constraints. This is the first place in the guide that a theoretical assumption is explained and the appropriate module manipulation is described in detail. The results of incrementing packages places high significance on the visual resolution dimension. The manual suggests exiting to the main menu and reentering the Constrain Fidelity Dimension. Once inside, the user follows the procedure to constrain a dimension level and returns to the package display option. The incremental package function now lists a display package that does not include the constrained dimension level. The money previously assigned to the fidelity dimension level recently constrained can now be used for other system improvements.

This display is helpful in determining which dimension level is changing as the packages are incremented or decremented. An asterisk appears to indicate which fidelity dimension has recently

changed. Since the optimal device design packages change by one level increments, this presentation method is good. The ability to set a cost constraint to see which package comes closest to approaching the budgeted cost is also helpful.

The fourth and final screen to be seen in the FOM module is labeled User Defined Packages. With this option it is possible to represent one or more user defined devices as points on the Graph Display of optimal TD designs. Selecting the yellow areas next to dimension levels increases the level of dimension by one. This user defined device can be displayed on the Graphic Display as a **P**. Any user defined device will end up below the benefit curve. This option allows the user to customize a training device based on his personal knowledge, making the system flexible.

The user defined packages can be saved in the same manner as those from the optimal cost to benefit curve. A small blue insert appears and gives clear instruction of how to save the package.

The last function of the FOM is called Save Changes. This option is only carried out if changes were made in the task cluster or a set of constrained fidelity dimensions would need to be changed. If no changes had been made then this option need not be exercised. The training device designs have already been saved within the module.

TRAINING DEVICE SELECTION MODULE

In the Training Device Selection Module (TDSM), OSBATS shifts from designing training devices to integrating individual training devices into a whole training system. This integration includes an analysis of existing devices, classroom training, and the actual equipment as parameters in system design. The guide does not adequately prepare the user for the change of focus. The module seems to continue the design of optimal training device when it has already accomplished this task. The documentation could make it clear that if individual training device design is the goal of the user, then model operations are complete. The training devices produced in FOM are designed as precisely as the model is capable. Further model analysis focuses on system designs.

The TDSM integrates individual training devices into a training system. The TDSM applies cost-effectiveness analysis to select the training device that meets training requirements for all tasks at the minimum cost. The TDSM assumes that cost is a simple linear function of training device usage. The module then allocates tasks to training devices within the system based on the device that produces the best level of performance given the cost. The TDSM minimizes the cost to train each task on available training devices using learning curve data.

The output of the TDSM is the aggregate use of each training device across all tasks. The TDSM recommends the device that should be used to train each task, the level of training at which each device should be employed, projected life-cycle cost of training, and the expected performance level of each selected device.

The TDSM is designed to inform the user of how many devices need to be purchased based on a training system utilization plan. Before operation of the module, three inputs must exist: a created PMS or FMS task cluster, an ifout package, and a set of new training device designs. The default task cluster must be in use by the system.

Once the TDSM has been selected, four options are available:

- Examine/Modify Task Set
- Show/Edit Training Devices
- Display Results
- Save Changes

In the first option, Examine/Edit Task Set, the display format is identical to the FOM edit/examine task set option. Tasks are excluded according to user specifications. For this module, all tasks should remain in the analysis, although exclusion is still possible. It is not clear why tasks would need to be edited since the training device has already been designed.

The second option, Show/Edit Training Devices, presents all existing and prototype training devices and includes actual equipment and classroom training. The presentation format is the same as the previous option. All devices can be included in the analysis or excluded according to user requirements. Training devices developed from the FOM can be deleted from the examination as well as excluded. Those existing devices imputed during data collection can only be excluded, not deleted. The presentation format is clear; users will have no difficulty understanding that excluded tasks will not be included in the modules results section. The third main menu option, Display Results allocates the training of tasks in the default task cluster to the devices. Within this option are four different selections:

- Results Summary
    - reiterate
- Graph This
- Results by Task
- Device Data

Current Task Cluster : default
Overall Result Display

Training Devices	Tasks Trained of 38	Cost to Train (000's)	Training Hours / Student	Dev. Rqd.	Hours / Year / Device	Assumed Hours / Dev/Yr
Classroom Training	4	0.19	5.97	3	1593	2000
Actual Equipment	34	51.73	59.80	20	2392	2500
Totals		51.92	65.77			
Number of Students / Year				: 800		
Number of Tasks NOT trained to standard:				0		

Save Results	Graph This	Comment	Print
Re-iteration	Results by Task	Help	Save Scr
Restore Res.	Device Data	Model Menu	Toggle Scr

Chart 2 Overall Result Display

The first screen display option is the overall result. The summary table, as shown in Chart 2, presents a matrix that reports six items:

- the number of task trained by each training device
- the number training hours device is used per student
- the number of devices needed to provide criterion training
- the total hourly use per device
- estimates annual device utilization
- number of tasks left untrained.

The matrix also provides the cost of the system given training device cost and required number of devices purchased. The table should be restructured to convey more effectively the importance of the values. The cost value is not highlighted or placed in a position of screen importance. The two variables that appear outside of the matrix (and appear important) are the number of students/year and the number of tasks NOT trained to standard. These values can not be manipulated within the module and are informative only.

Other values that hold positions of importance in the matrix display are Cost to Train and Hours of training required. These are based on the number of students trained. No method of changing the number of students to be trained is possible within the running of OSBATS. (OSBATS must be exited to change this value.) What is done about tasks that are not trained on any devices? Since the example data does not contain tasks that cannot be trained, no explanation of how to view these tasks is provided. Will the tasks be presented in the Results By Task option, or since they have no assigned training device, will they not be displayed? The documentation does not address this area.

The user's guide instructs users to save this system package of training devices after the first screen. The optimal training system saved is the result of TDSM. Users are told to save a package as optimal before any reasons are given as to why this is considered the optimal mix of devices. No further screens are discussed. This lack of discussion gives the impression that the remaining module screens are not needed for effective evaluation.

The TDSM operates with the full task cluster (default). The reasons why the default task cluster is needed in this module is not explained. It may be inferred that training should consist of addressing all tasks within a domain, even if they are not considered trainable. However, the user may become uncomfortable unless more specific explanation about why task clusters are being changed is given.

Within the overall summary screen, there is an option to Reiterate. Reiteration is used when a large discrepancy exists between Assumed hours/Device/year and actual Hours/Device/year. The calculation for the per-student cost and per-student time to train the tasks depends on the estimate of annual device use. Reiteration replaces the assumed hours/device/year with the actual figure; the values dependent on these estimates are then recalculated. The module treats the actual hours/device/year figure as if it were a fact, when it is only an estimate. With the Reiteration option, the user is simply given an option of which figure to use. The complex definition and presentation of Reiteration is too confusing to be used well.

The value of the overall results screen is that the user can change the devices in the analysis. The results are quick and fairly easy to understand, but the process of changing the devices requires too many steps. The user must first exit the summary screen, return to the main module menu, and select an alternate menu option Show/Edit Training Devices. Training device inclusion and exclusion must be performed before returning to the model selection menu to select the Display Results option. In order to manipulate training devices for the analysis requires four mouse click interactions, plus one mouse click for each training device that is reclassified. By the time the user reestablishes the system grouping he may forget what he was trying to accomplish.

The second option Graph This allows quick assessment of the hours and cost allocated to each device. The cost values are presumably a total of all devices in the system. This information is not clear when the graph is displayed. The Graph This icon appears in the middle row of module selection options. Typically, this indicates that the option could be selected from any of the other module screens; however, the Graph This option can only be selected when the overall results screen is being viewed. Thus, Graph This is an option beneath the overall results screen and not an module option. This indicates that a problem associated with the hierarchial structure may exist.

The third option is Results by Task. Within this option the module hierarchy system becomes confusing again. The options to view individual task data can only be accessed when the Results by Task option is on the screen. Graphing of the task data is only possible from the individual task data screen. The module levels are not transparent. The bottom seven lines, which are intended to portray system options, are continually changing. It is difficult to determine if all module screens have been viewed.

There are several problems in the presentation of the graphs. First, the x-axis continually changes its scale. The x-axis is a time scale and the end point changes with each task, making it difficult to visually compare graphs. The second problem is in the presentation of the entry level performance. Entry level performance is displayed as a single broken line that travels across the entire time scale. If training is taking place and time is a dynamic variable, then entry performance would be single point not a line.

The third problem concerns the graphical representation of a training task that requires more than one training device. Documentation does not cover the following system conventions, thus information in this paragraph is based on the author's observations. The module labels each training device with a number. Each task is assigned the same number as the training device that trains that task. In the graph, the top line indicating criterion performance is always associated with the

training device that completes criterion training. If more then one training device exists, then training must transfer from one training device to the other at a given point in training. The point on the graph where this occurs is represented by the reference number of the first training device. The position of the number indicates how much training time the task requires before a different device in needed to complete training. The number is positioned vertically to indicate how much training (training time) has occurred prior to transfer.

If the graphs of individual task data were not parallel lines displaying entry and criterion performance levels, but rather a curved line indicating where entry performance began and when criterion performance was reached, more information could be obtained from the graph (assuming x-axis stabilization). For example, the slope of the line would help to indicate task difficulty and a point on the line or a vertical line could indicate where transfer from one device to the next would be optimal. The present method of presentation does not make use of the inherent advantages to displaying information graphically.

The fourth option, Device Data, displays the data used to determine the cost of each training device. The cost categories are standard measures: investment cost, life cycle of device, hourly investment cost, annual fixed operating cost, variable operating cost, and total hourly cost. This display is not interactive; it displays data base information and estimates obtained from the prototype training device designed by the model. The display does not list how many devices the cost values reflect.

The last operation to be conducted is to save any changes that were made in the task cluster or the training device set (deleted a training device). The system prompts the user for a name and quickly saves the packages. This completes the Training Device Selection Module. Ideally, an optimal system configuration has been established to be refined in the Resource Allocation Module (RESAM).

RESOURCE ALLOCATION MODULE

The Resource Allocation Module (RESAM) provides results similar to those of the TDSM. However, the methods differ in two respects: (1) the TDSM assumes cost as a linear function of training device usage, while the RESAM breaks cost down into a number of linear segments, (2) the RESAM allows the user to specify constraints on training device use. Constraints may specify the maximum time that a device may be used or the minimum performance level for which a training device may be employed on a specific task. The RESAM involves great computational complexity and calculation time may be ten times longer than in the TDSM. This increased computational load limits the amount of user interaction possible.

The RESAM begins at the Problem Solution Selection Menu, not at the modules main menu level. Prior to this module, entry into a module resulted in the listing of the module's main menu. The Problem Selection Menu is a listing of the system packages designed in the TDSM. Once a package has been selected, the program should return to the module's main menu; this does not always happen. In one out of three occasions the program does not return to the main menu, but stays in the program selection menu after confirmation of selection. A mouse click on a selection labeled Return is required. There is no consistent method adopted by the module.

Once the package to be optimized is chosen, the main menu will appear. There are five options within the RESAM:

- Select Problem
- Show/Edit Constraints
- Optimize Allocation
- Save Constraints
- Module Selection Menu

Before operation of this module, four inputs must already exist: (a) a created FMS task cluster from the SCM, (b) a saved ifout package from the IFSM, (c) saved candidate training devices as designed in the FOM, and (d) a saved training system package from the TDSM.

The first step after a package has been selected from the Select Problem option is to run the Optimize Allocation option in the main menu. The Select Problem option need not be chosen because this is the display that the RESAM enters from the module selection menu. Once the Optimize Allocation option has been chosen, the

model immediately begins computations and a displayed message informs the user that up to twenty minutes are required for optimization. Other selections available in this option are:

- Overall Results
- Graph
- Results by Task
  - Individual Task Data
  - Graph

After computations are completed, the module enters the overall results display. This display is nearly identical to the overall results display found in the TDSM, and contains the same inconsistencies. The summary table presents a matrix that reports six items:

- the number of task trained by each training device
- the number training hours device is used per student
- the number of devices needed to provide criterion training
- the total hourly use per device
- estimates annual device utilization
- number of tasks left untrained.

The matrix also provides the cost of the system given training device cost and required number of devices purchased. The table should be restructured to convey more effectively the importance of the values. The cost value is not highlighted or placed in a position of screen importance. The two variables that appear outside of the matrix (and appear important) are the number of students/year and the number of tasks NOT trained to standard. These values can not be manipulated within the module and are informative only.

The RESAM's overall results display does not give the user the option to reiterate. The utilization hours estimates are not changeable. The module's menu allows a graphical representation of this information from the overall results display. Two vertical bar graphs appear which shows total cost associated with each device and the number of hours of device usage.

The results by task submenu option displays each training task and the devices used to train the task. The display also gives information of how many hours of training are spent on each device per task. This information is very valuable. OSBATS has allocated specific devices for specific tasks and estimated the time needed to train the task to criterion. This information can be used to design the program of instruction in great detail. The documentation does not convey the potential impact of this display. The documentation describes only the content of the display, not the possible applications of the content.

From the Results by Task display, the next display to be viewed is the Individual Task Data. Each task is displayed in a matrix format which conveys the following information

- training standard and entrance performance
- allocation of training devices to task
- cost to train each student
- criterion training performance required

The individual task data display gives detailed information of how tasks and training devices are allocated to each other and the cost of this training configuration. From within this display, a graphical representation of the information is possible with a Graph selection option. The graph display format is identical to one found in the TDSM and contains the same difficulties.

The module has a presentation element that may be difficult for users to understand. When a training system package is optimized, sometimes the problem is too large for the model (requires additional memory). The program informs a user that module computations must be done off-line. Two information screens instruct users to exit the module and complete computations off-line. This is an unexpected difficulty in the programs operation. The process of computation and reentering OSBATS RESAM can take over fifteen minutes. This is a long down time for a system user. The user may not feel comfortable with exiting the program.
The last option to be exercised in the RESAM is the Save Constraints option. The user's manual does not document how or why this option should be used. When it is selected, the display prompts the user for a name. Where this will appear is not specified. Although through trial and error it was determined that it appears in the package selection menu as a new option.

APPENDIX B

TASK LIST FOR GUNNERY POSITIONS

VIGS AND TOPGUN

APPENDIX B

Gunnery Task List

The task list below has been adopted from Morrison and Hoffman (1987). The column labeled "Device" indicates that training of corresponding task is possible (yes) or not possible (no) on the training device listed. The label "trainable" in this context means that enough cues exist in the training device to train the task.

Activity	Options	Devices VIGS	Devices TOPGUN
1. Prepare Station for Operation (PREOPS)		No	No
2. Prepare to Fire Checks (PRE-FIRE)		No	No
	2.1 Prepare for Offensive		
	2.2 Prepare for Defensive		
3. Acquire Targets			
	3.1 Search for Targets	Yes	Yes
	3.1.1 Search Open Hatch - Day		
	3.1.2 Search Open Hatch - Night		
	3.2 Detect/Locate/Identify	No	No
	3.3 Evaluate Situations	No	No

Activity	Options	Devices VIGS	Devices TOPGUN
4. Engage Single Target with Main Gun		Yes	Yes
	4.1 Engage, Offensive, Precision Gunnery	Yes	No
	4.2 Engage, Defensive, Precision Gunnery	Yes	No
	4.3 Gunner Can't Identify Target	No	No
	4.4 Engage Using TIS	Yes	Yes
5. Adjust Fire			
	5.1 Use Reengage Technique	No	No
	5.2 Use Standard Adjustment	Yes	Yes
	5.3 Use TC Adjustment	No	No
6. Engage Single Target with COAX		Yes	No
7. Engage Multiple Targets with Main Gun		Yes	Yes
8. Engage Targets with CAL .50			
	8.1 Simultaneous Targets	Yes	No
	8.2 CAL. 50 Targets	Yes	No
9. Engage Targets Using Degraded Gunnery Techniques			
	9.1 Battle Sight Gunnery	No	No
	9.2. Ineffective LFR	Yes	Yes
	9.3 Multiple Returns for LFR	No	No
	9.4 No Range Display (Loss of Symbology)	No	No
	9.5 Crosswind Sensor Failure	No	Yes
	9.6 Cant Sensor Failure	No	Yes
	9.7 Lead Angle Sensor Failure	No	Yes

Activity	Options	Devices VIGS	Devices TOPGUN
	9.8 GPS Failure (Day Channel)	Yes	Yes
	9.9 Gunners Auxiliary sight	Yes	Yes
	9.10 GPS/TIS System Failure	Yes	Yes
	9.11 Stabilization System Failure	No	No
	9.12 Loss of Turret Power (Manual)	No	No
10. Engage Target(s) from TC Position		No	No
11. Assess Results of Engagement		No	No

APPENDIX C

TASK LIST FOR OSBATS APPLICATION

TANK GUNNERY

APPENDIX C

Task List
OSBATS Application

The following task list was used as input for running the OSBATS model. The list was modified from Morrison and Hoffman (1987). A shortened task list was adopted to lessen data collection and input time. The list is a cross section of tasks that were both trainable and non-trainable on the two candidate tracing devices, VIGS, TOPGUN. The label "trainable" indicates that enough cues are present in the candidate devices to train the task.

Tasks Determined Non- trainable on candidate devices:

1001. Prepare Station for Operation (PREOPS)

1002. Prepare to Fire Checks (PRE-FIRE)

1003. Degraded Gunnery Techniques II

Tasks Determined Trainable on candidate devices:

1004. Acquire Targets

1005. Adjust Fire

1006. Engage Multiple Targets with Main Gun

1007. Engage Single Target with Main Gun

1008 Degraded Gunnery Techniques I

APPENDIX D

DATA BASE FOR OSBATS APPLICATION

TANK GUNNERY

\OSBATS\DATA\TASKDATA.INP

```
1001  PREOPS
1002  PREFIRE
1003  DEG_GUN2
1004  AQ_TRGT
1005  ADJ_FIRE
1006  MULT_ENG
1007  SINGL_ENG
1008  DEG_GUN1
```

This file contains the numbers and short names for the task set.

\OSBATS\DATA\LONGDISC.INP

```
PREPARE STATION FOR OPERATIONS
PREPARE STATION FOR FIRING
ENGAGE DEGRADED GUNNERY GROUP 2
ACQUIRE TARGETS
ADJUST FIRE
ENGAGE MULTIPLE TARGETS - MAIN GUN
ENGAGE SINGLE TARGET - MAIN GUN
DEGRADED GUNNERY - GROUP 1
```

This file contains the long descriptions or names of the tasks.

\OSBATS\DATA\LPOINTS.INP

.2500	.75
.2500	.75
.0000	.75
.2500	.75
.2500	.75
.0000	.75
.0000	.75
.0000	.75

The learning points file contains estimates of the average initial ability of students to perform the task, and the desired exit performance level. The scale is 0 to 1.0 with ordinal levels assigned to decimal values.

\OSBATS\DATA\EQPHRS.INP

12.00	1.50	8.00	8.00	.00
4.00	.00	4.00	4.00	3.00
3.00	6.00	12.00	8.00	5.00
20.00	.00	14.00	8.00	.20
.20	.30	.00	1.00	5.00
.00	.00	18.00	12.00	11.00
2.00	.00	18.00	12.00	11.00
5.50	6.00	18.00	12.00	11.00

The equipment hours file contains cost and time-to-train information about the tasks. The first column is classroom hours to train the task. The second through fourth columns concern actual equipment hours needed to train the tasks in the absence of a training device. The second column is non-operational hours, the third is operational time, the third column provides required setup time on the actual equipment. The last column covers cost (in thousands) of other equipment needed to train using the actual equipment.

\OSBATS\DATA\SIMDET.INP

```
0  0  1  0  1  0
0  0  0  1  0  0
0  1  1  1  1  1
0  1  0  1  0  1
0  0  0  1  1  1
0  1  0  1  1  1
0  1  0  1  1  1
0  1  1  1  1  1
```

This file provides the input for the indications for simulation calculations in the Simulation Configuration model. The ones (1) in the columns indicate special requirements for training devices such as special weather (second column), special situations (third), special equipment (third), training effects (fifth), and time savings (sixth). The first column represents a judgment that simulation is an absolute requirement for training, e.g. due to safety considerations, etc.

\OSBATS\DATA\IF_NAME.INP

adaptive	1.59	253.0
adjn_cai	1.70	236.0
aug_cues	1.30	155.0
aug_fdbk	1.30	97.0
autodemo	2.16	54.0
coaching	1.10	156.0
crash_ov	4.77	29.0
flt_frez	3.80	35.0
graph_rp	2.70	76.0
hardcopy	.00	.0
init_cnd	4.30	98.0
ios_dspl	.00	.0
parm_frz	2.95	35.0
perf_ind	1.40	51.0
perfalrt	1.30	61.0
perfmeas	1.40	215.0
pos_frez	3.30	35.0
proc_mod	3.40	74.0
realtime	3.70	137.0
rec/play	1.42	92.0
reset/re	4.63	30.0
scen_ctl	4.40	155.0
sys_frez	3.10	24.0

This file provides the short names, the benefit values, and the estimated costs of the instructional features.

\OSBATS\DATA\LONGIF.INP

Automated Adaptive Training
Adjunct Computer Assisted Instruction
Augmented Cues
Augmented Feedback
Automated Simulator Demonstration
Automated Cueing and Coaching
Crash Override
Flight System Freeze
Remote Graphics Replay
paper copy of training results
Initial Conditions
instructor/operator station display
Parameter Freeze
Performance Indicators
Automated Performance Alerts
Automated Performance Measurements
Positional Freeze
Procedures Monitoring
Real Time Simulation Variables Control
Record / Replay
Reset / Reposition
Scenario Control
Total System Freeze

This file contains the long, or full, names of the instructional features.

\OSBATS\DATA\TABLE9.INP

1001	1	1	0	0	0	1	0	0	0	1	0	0	0	1	0	1	1	0	1	1	0
1002	1	1	0	0	0	1	0	0	0	1	0	0	0	1	0	1	1	0	1	1	0
1003	1	1	0	0	0	1	0	0	1	1	0	0	0	1	0	1	1	1	1	1	1
1004	1	1	1	1	0	0	0	0	1	1	0	1	0	1	0	0	1	1	0	1	1
1005	1	1	1	0	0	1	0	0	1	1	0	0	0	1	0	1	1	1	1	1	1
1006	1	1	1	0	0	1	0	0	1	1	0	0	0	1	0	1	1	1	1	1	1
1007	1	1	1	0	0	1	0	0	1	1	0	0	0	1	0	1	1	1	1	1	1
1008	1	1	0	0	0	1	0	0	1	1	0	0	0	1	0	1	1	1	1	1	1

This file is generated by the instructional feature rule set, and identifies (with ones in the columns) the instructional features applicable for the tasks (identified by number in the first column. The order for the instructional features must match the order of features in the other files.

\OSBATS\DATA\DIMTEXT.INP

```
none
Cultural Lights
Add Wpns Blast
Add Damaged Veh
Add Airbrn Veh
Add Mvng Grnd Veh
|
  5 x  5 km
 10 x 10 km
 10 x 20 km
 10 x 30 km
 20 x 30 km
 30 x 30 km
 30 x 40 km
|
none
3 D. F.
5 D. F.
6 D. F.
|
none
smoke and dust
rotor wash
|
Stationary
Seat Shaker
Add G-Seat
|
none
Wpns,Skid,Fail
Add nor opt nse
Add abnor opt nse
|
Plane w/ Trees
Add Genric Featrs
Realistic Densty
Low Dnsty Hydro
Med Dnsty Hydro
Hgh Dnsty Hydro
|
40 x 40 Deg
40 x 50 Deg
40 x 60 Deg
|
m2 at 0.3 km
m2 at 0.5 km
m2 at 1.0 km
m2 at 2.0 km
m2 at 3.0 km
m2 at 4.0 km
|
40 x 40 Deg
40 x 50 Deg
50 x 50 Deg
50 x 60 Deg
40 x 50 Deg
40 x 60 Deg
50 x 60 Deg
|
Lines+Polygons
Modulatg Fnctns
Few Digit Ph
More Digit Ph
Many Digit Ph
```

This file describes the levels of each fidelity dimension. The file actually is a single string, in two columns here for ease of presentation. Again the order must match the other fidelity files.

\OSBATS\DATA\TECH.INP

0.0	0.45	0.9	-1.0	-1.0	-1.0	-1.0
0.5	0.58	0.62	0.66	0.74	0.82	0.9
0.0	0.36	0.63	0.9	-1.0	-1.0	-1.0
0.0	0.09	0.18	0.36	0.63	0.9	-1.0
0.0	0.36	0.9	-1.0	-1.0	-1.0	-1.0
0.0	0.27	0.54	0.9	-1.0	-1.0	-1.0
0.5	0.58	0.66	0.74	0.82	0.9	-1.0
0.77	0.86	0.94	-1.0	-1.0	-1.0	-1.0
0.52	0.71	0.86	0.93	0.95	0.96	-1.0
0.46	0.51	0.57	0.62	0.72	0.79	0.88
0.5	0.62	0.74	0.84	0.9	-1.0	-1.0

This file provides the technical performance levels for the fidelity dimensions, and again the order of rows must match the order of fidelity dimensions in the other files.

\OSBATS\DATA\MINMAX.INP

.0	336.0	.5	.5
100.0	3360.0	.5	.0
.0	3024.0	.1	.6
.0	504.0	.6	.4
.0	192.0	.4	.8
.0	192.0	.8	.4
80.0	2520.0	.4	.0
20.0	672.0	.9	.0
50.0	2016.0	.6	.0
10.0	336.0	.2	.0
75.0	1344.0	.5	.0

This file contains the estimated minimum (first column) and maximum (second column) costs for the development of the fidelity dimensions. The intermediate costs of fidelity levels are calculated using the exponent (third column) and minimum weight(last column).

\OSBATS\DATA\CUERSP.INP

```
1001 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
1002 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
1003 0.45 0.50 0.00 0.64 0.36 0.00 0.50 0.77 0.95 0.64 0.90
1004 0.45 0.50 0.00 0.00 0.00 0.00 0.50 0.77 0.95 0.00 0.50
1005 0.45 0.50 0.00 0.64 0.36 0.00 0.50 0.77 0.93 0.64 0.90
1006 0.45 0.50 0.00 0.64 0.36 0.00 0.50 0.77 0.93 0.64 0.90
1007 0.45 0.50 0.00 0.64 0.36 0.00 0.50 0.77 0.93 0.64 0.90
1008 0.45 0.50 0.00 0.64 0.00 0.00 0.50 0.77 0.93 0.64 0.90
```

The four digit number in the first column is the task number used at the school. The following numbers refer to the fidelity level and associated normalized benefit required to "PERFORM" the task. This file is generated from the Fidelity rule base in response to task information. The order of the columns is critical.

\OSBATS\DATA\DEV_NAME.INP

M1_TANK
TOPGUN
VIGS

These are the short names for the training devices that already exist at the school. The devices are typically included in the Training Device Selection and Resource Allocation models.
*
\OSBATS\DATA\LONGDEV.INP

M-1 TANK
TOPGUN PRECISION GUNNERY TRAINER
VIDEODISC GUNNERY SIMULATOR

These are the long, or full, names for the training devices.

\OSBATS\DATA\DEVFID.INP

```
1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00
.45   .58   .00   .90   .00   .27   .66   .77   .95   .00   .62
.45   .50   .00   .90   .00   .27   .66   .77   .86   .00   .62
```

These are the fidelity level values for the devices that already exist at the school. The values come from a fidelity analysis,and allow the models to equate the existing devices to the device concepts developed through the use of the model. The order of the information is critical for correct use by the model. The first line refers to the first device listed in DEV_NAME.INP and LONGDEV.INP. The order of values must match the order of fidelity dimensions contained in FDIMENS.INP, DIMDESC.INP, and DIMTEXT.INP.

OSBATS\DATA\DEV_CST.INP

5000.0	100.000	.300	15	2000
750.0	1.850	.010	13	2000
1300.0	4.250	.010	12	2000

These values concern the costs associated with the existing devices. The first value is the development cost. The second column contains yearly fixed costs. The third column is the hourly variable costs (rather than normal fixed costs for operations). The fourth column has the life cycle time in years. The fifth column contains the estimated (or required) hours of operation per year. These values are used in the Training Device Selection and Resource Allocation models.

\OSBATS\DATA\DEV_IF.INP

0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0	0	1	0	1	1	1	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0

This file describes the instructional features present on theexisting devices. Again the order of the columns must agree with the order of instructional features as given in IF_NAME.INP,LONGIF.INP, and elsewhere in the model data.

\OSBATS\DATA\SENVARS.INP

```
0.70  1000.0   3.00  0.1  3000.00  1.0   80000.00   0.60   10.00
```

This file contains special model information used for calculating learning curves, setting limits on tradeoff routines, and setting initial values for weights used in calculations (that can be adjusted by the user).

\OSBATS\DATA\FDIMENS.INP

```
Area_Effects
Map_Size
Platform_Mot.
Point_Effects
Seat_Motion
Sound_Effects
Visual_Content
Visual_Front
Visual_Resol.
Visual_Side
Visual_Texture
```

This file contains the short names of the fidelity dimensions.

\OSBATS\DATA\DIMDESC.INP

```
Background scene content in visuals
Size of the Visual operations area
Degrees of motions of simulator platform
Moving elements in visual scenes
Seat force cuing devices
Special sound effects & operating noises
Visual display content
Forward visible field of view
Resolution capability of visual display
Size of visual display to side
Visual content texture or realism
```

This file provides descriptions of the fidelity dimensions.

APPENDIX E

DATA BASE FOR ASTAR APPLICATION

TANK GUNNERY

APPENDIX E

DATA BASE FOR ASTAR APPLICATION

TANK GUNNERY

Task list
ASTAR Application

(data base information)

The task list below was adopted from Morrison and Hoffman (1987). The list was used as task and subtask inputs for application with the ASTAR training device design aid.

TOP GUN (Gunnery Trainer)

1.0000 Prepare Station for Operation (PREOPS)

2.0000 Perform Prepare-To-Fire (PRE-FIRE) Checks
- 2.1000 Prepare for Offense Deployment (moving)
- 2.2000 Prepare for Defensive Deployment (stationary)

3.0000 Acquire Target
- 3.1000 Search For Target(s)
  - 3.1100 Search Open Hatch--Day
  - 3.1200 Search Closed Hatch--Day
  - 3.1300 Search at Night
- 3.2000 Detect/Locate/Identify Friend or Foe (IFFN)

4.0000 Engage Single Target with Main Gun
- 4.2000 Engage from Defensive Posture, using precision gunnery
- 4.3000 Engage Targets with Thermal Imaging System (TIS)
- 4.4000 Engage from Offensive Posture, using precision gunnery

5.0000 Adjust Fire (after a miss)
- 5.1000 Use Reengage Technique
- 5.2000 Use Standard Adjustment
- 5.3000 Use TC Adjustment

6.0000 Engage Single Target With the COAX

7.0000 Engage Multiple Targets With the Main Gun

8.0000 Engage Targets using Degraded Gunnery Techniques

9.0000 Engage Targets from the TC Position

10.000 Assess Results of the Engagement

ASTAR Application
Description of Equipment

(data base information)

The following is the data base for displays (d) and controls (c) used in the gunners position on the actual equipment. This list of displays and controls is for ASTAR Level 3 analysis.

Actual Equipment:

c1 Gunner Power Control Handle
- c1.1 laser button
- c1.2 palm switch
- c1.3 trigger

c2 Thermal Imaging System Control and Adjustments

c3 Laser Range Finder Select (ARM 1stRTN, LAST RTN, SAFE)

c4 Magnification Select (3x, 10x)

c5 Gunners Primary Sight Panel (Normal, Thermal, NBC)

c6 Gun Select (Main, COAX)

c7 Ammo Select (Heat, Sabot, Hep, BH)

d1 Gunners Primary Sight (GPS)

d2 Gunners Auxiliary Sight (GAS)

d3 Gunners Reticile

APPENDIX F

EXPERT SYSTEM TASK QUESTIONS

APPENDIX F

EXPERT SYSTEM TASK QUESTIONS

This appendix illustrates the specific questions that the expert system requires to build its data base about task information. The questions are presented in the sequence asked by the expert system, with the number of the answer listed below. The information for the first two tasks was not reported since the last 5 tasks were able to illustrate all the systems questions.

==============================
==============================

Please input the TASK NUMBER of the current task (four digit maximum)
:1003

Performance of the task _____ an out-of-cockpit view
1. requires
2. does not require

1

When judging distances and ranges, the most convenient unit of measure is
1. feet
2. meters

2

Performing the task requires
1. absolute judgement of altitude in feet
2. absolute judgement of slant range in meters
3. judgement of clearance
4. detection of targets or navigation waypoints/checkpoints
5. slope landing
6. none of the above activities

3 4

The task requires judgement of
1. vertical clearance
2. lateral clearance
3. neither vertical nor lateral clearance

1 2

Objects that must be detected to perform this task include
1. tanks
2. APC's
3. 5-ton trucks
4. 2.5-ton trucks
5. 4-wheeled vehicles (e.g. jeep)
6. other aircraft
7. troops
8. other objects

1 2 3 4 5 6 7 8

Please input the maximum likely RANGE, in meters, at which the tank must be detected
: 5000
Please input the maximum likely RANGE, in meters, at which the APC must be detected
: 5000

Please input the maximum likely RANGE, in meters, at which the five-ton truck must be detected
: 5000

Please input the maximum likely RANGE , in meters, at which the 2.5 ton truck must be detected
: 5000

Please input the maximum likely RANGE, in meters, at which the four-wheeled vehicle must be detected
: 5000

Please input the maximum likely RANGE, in meters, at which the other aircraft must be detected
: 5000

Please input the maximum likely RANGE, in meters, at which the troops must be detected
: 2000

Please input the maximum likely RANGE, in meters, at which the other aircraft must be detected
: 5000

Please input the maximum likely RANGE, in meters, at which the troops must be detected
: 2000

Please input the minimum size dimension, in feet, of the other object. (Consider the object to be the object most difficult to detect at the range it must be detected.)
: 3

The minimal scene content required for effective training for this task is
1. flat textured ground plane with scattered trees
2. level 1 plus paved landing field, unimproved confined landing area, and generic terrain relief
3. level 2 plus realistic density/configuration of trees
4. level 3 plus low density hydrographic and cultural features (desert terrain)
5. level 4 plus medium density hydrographic and cultural features (countryside)
6. level 5 plus high density hydrographic and cultural features (urban areas)

1

The minimum size of the topographical data base required by this task is
1. 5 x 5 km or smaller
2. 10 x 10 km
3. 10 x 20 km
4. 10 x 30 km
5. 20 x 30 km
6. 30 x 30 km
7. 30 x 40 km

1

The environment in which the task is performed is
1. airport environment
2. desert, vater, or other visually simple environment
3. countryside, forest, or visually similar environment
4. urban or other visually complex environment

2 3 4

Performance of the task requires
1. peripheral cues to provide "flow" information for judging velocity and acceleration for alerting the pilot
2. operation of the aircraft when severely pitched up or rolled
3. identification of ground locations, such as landing fields, confined areas, targets, threats, and navigation waypoints/checkpoints
4. none of the above conditions

3

Ground locations that must be identified to perform this task are viewed
1. through the center 50% of the front window
2. through the center 75% of the front window
3. anywhere through the front window
4. through either the front or side windows

1

The content elements that would enhance the effectiveness of training on the task include
1. cultural lights
2. weapons blast
3. damaged vehicles
4. airborne vehicles
5. moving ground vehicles
6. smoke and dust
7. rotor wash
8. none of the above

2 3 5 6

Specific cues for performance of the task are provided by the following kinds of motion:
1. roll
2. pitch
3. yaw
4. vertical acceleration
5. lateral acceleration
6. longitudinal acceleration
7. there are no specific motion cues

7

The type of activity involved in the task is
1. continuous control movement
2. procedures
3. perception
4. decision making/rule using
5. using symbolic information
6. voice communication

1 2 3 4 6

The following audio signals (other than voice) provide cues for performance during the normal operation of this task:

1. correlated rotor flap
2. correlated engine noise (normal operation)
3. correlated engine noise (abnormal conditions such as engine failure, overspeed, and underspeed)
4. weapons firing
5. low/high RPM warning signal
6. skid noise (contact of skid with ground)
7. drive shaft/clutch failure
8. compressor stall
9. other audio signal
10. none of the above

4

Of the discrete audio signals (weapons firing, low RPM warning, skid noise, or clutch failure) that normally occur during the performance of this task, _____, are correlated with visual cues.

1. All
2. Some
3. None

1

The task _____ the performance of an emergency procedure

1. requires
2. does not require

1

A discrete audio signal (weapon firing, low RPM warning, skid noise, or clutch failure) _____ for the initiation of the emergency procedure

1. provides a cue
2. does not provide a cue

2

Please input the TASK NUMBER of the current task (four digit maximum)
: 1004

Performance of the task ___ an out-of-cockpit view
1. requires
2. does not require

1

When judging distances and ranges, the most convenient unit of measure is
1. feet
2. meters

2

Performing the task requires
1. absolute judgement of altitude in feet
2. absolute judgement of slant range in meters
3. judgement of clearance
4. detection of targets or navigation waypoints/checkpoints
5. slope landing
6. none of the above activities

3 4

The task requires judgments of
1. vertical clearance
2. lateral clearance
3. neither vertical nor lateral clearance

1 2

Objects that must be detected to perform this task include
1. tanks
2. APC's
3. 5-ton trucks
4. 2.5 ton trucks
5. 4 wheeled vehicles (e.g. Jeep)
6. other aircraft
7. troops
8. other objects

1 2 3 4 5 6 7 8

The type of activity involved in the task is
1. continuous control movement
2. procedures
3. perception
4. decision making/rule using
5. using symbolic information
6. voice communication

2 3 4 5 6

Please input the TASK NUMBER of the current task (four digit maximum): 1005

Input the text for the variable data:
Performing task requires
1. absolute judgement of altitude in feet
2. absolute judgement of slant range in meters
3. judgement of clearance
4. detection of targets or navigation waypoints/checkpoints
5. slope landing
6. none of the above activities

3 4

The task require judgement of
1. Vertical clearance
2. lateral clearance
3. neither vertical nor lateral clearance

1 2

Objects that must be detected to perform this task include
1. tanks
2. APC's
3. 5 - ton trucks
4. 2.5 - ton trucks
5. 4 - wheeled vehicles (e.g. Jeep)
6. other aircraft
7. troops
8. other objects

1 2 3 4 5 6 7 8

Please input the maximum likely RANGE, in meters, at which the tank must be detected:
: 2500

Please input the maximum likely RANGE, in meters, at which the APC must be detected:
: 2500

Please input the maximum likely RANGE, in meters, at which the five-ton truck must be detected
: 2500

Please input the maximum likely RANGE, in meters, at which the five-ton truck must be detected
: 2500

Please input the maximum likely RANGE, in meters, at which the four-wheeled vehicle must be detected
: 2500

Please input the maximum likely RANGE, in meters, at which the other aircraft must be detected
: 2500

Please input the maximum likely RANGE, in meters, at which the troops must be detected
: 1800

Please input the maximum likely RANGE, in meters, at which the other object must be detected. (Consider the other object to be the most difficult object to detect at the range it must be detected.)
: 1800

Please input the minimum size dimension, in feet, of the other object. (Consider the other object to be the object most difficult to detect at the range it must be detected.)
: 3

The minimal scene content required for effective training for this task is
1. flat textured ground plane with scattered trees
2. level 1 plus paved landing field, unimproved confined landing area, and generic terrain relief
3. level 2 plus realistic density/configuration of trees
4. level 3 plus low density hydrographic and cultural features (desert terrain)
5. level 4 plus medium density hydrographic and cultural features (country side)
6. level 5 plus high density hydrographic and cultural features (urban areas)

1

The minimum size of the topographical data base is required by this task is
1. 5 x 5 km or smaller
2. 10 x 10 km
3. 10 x 20 km
4. 10 x 30 km
5. 20 x 30 km
6. 30 x 30 km
7. 30 x 40 km

1

Performance of the task requires
1. peripheral cues to provide "flow" information for judging velocity and acceleration or for alerting the pilot
2. operation of the aircraft when severely pitched up or rolled
3. identification of ground locations, such as landing fields, confined areas, targets, threats and navigation waypoints/checkpoints
4. none of the above conditions

3

Ground locations that must be identified to perform this task are viewed
1. through the center 50% of the front window
2. through the center 75% of the front window
3. anywhere through the front window
4. through either the front or side windows

1

The content elements that would enhance the effectiveness of training on the task include

1. cultural lights
2. weapons blast
3. damaged vehicles
4. airborne vehicles
5. moving ground vehicles
6. smoke and dust
7. rotor wash
8. none of the above

2 3 4 5 6

Specific cues for performance of the task are provided by the following kinds of motion:

1. roll
2. pitch
3. yaw
4. vertical acceleration
5. lateral acceleration
6. longitudinal acceleration
7. there are no specific motion cues

7

The following audio signals (other than voice) provide cues for performance during the normal operation of this task:

1. correlated rotor flap
2. correlated engine noise (normal operation)
3. correlated engine noise (abnormal conditions such as engine failure, overspeed, and underspeed)
4. weapons firing
5. low/high RPM warning signal
6. skid noise (contact of skid with ground)
7. drive shaft/clutch failure
8. compressor stall
9. other audio signal
10. none of the above

4

Of the discreet audio signals (weapons firing, low RPM warning, skid noise, or clutch failure) that normally occur during the performance of this task, ---------- are correlated with visual cues.

1. all
2. some
3. none

1

The task ______ the performance of an emergency procedure
1. requires
2. does not require

2

====================================
====================================

Please input the TASK NUMBER of the current task (four digit maximum)
: 1006

Performance of the task ___ an out-of-cockpit view
1. requires
2. does not require

1

When judging distances and ranges, the most convenient unit of measure is
1. feet
2. meters

2

Performing the task requires
1. absolute judgement of altitude in feet
2. absolute judgement of slant range in meters
3. judgement of clearance
4. detection of targets or navigation waypoints/checkpoints
5. slope landing
6. none of the above activities

3 4

The task requires judgments of
1. vertical clearance
2. lateral clearance
3. neither vertical nor lateral clearance

1 2

Objects that must be detected to perform this task include
1. tanks
2. APC's
3. 5-ton trucks
4. 2.5-ton trucks
5. 4-wheeled vehicles (e.g. Jeep)
6. other aircraft
7. troops
8. other objects

1 2 3 4 5 6 7 8

Please input the maximum likely RANGE, in meters, at which the tank must be detected
: 2500

Please input the maximum likely RANGE, in meters, at which the APC must be detected
: 2500

Please input the maximum likely RANGE, in meters, at which the five-ton truck must be detected
: 2500

Please input the maximum likely RANGE, in meters, at which the 2,5 ton truck must be detected
: 2500

Please input the maximum likely RANGE, in meters, at which the four-wheeled vehicle must be detected
: 2500

Please input the maximum likely RANGE, in meters, at which the other aircraft must be detected
: 2500

Please input the maximum likely RANGE, in meters, at which the troops must be detected
: 1800

Please input the minimum size dimension, in feet, of the other object. (Consider the other object to be the most difficult to detect at the range it must be detected.)
: 3
The minimal scene content required for the effective training for this task is

1. flat textured ground plane with scattered trees
2. level 1 plus paved landing field, unimproved confined landing area, and generic terrain relief
3. level 2 plus realistic density/configuration of trees
4. level 3 plus low density hydrographic and cultural features (desert terrain)
5. level 4 plus medium density hydrographic and cultural features (country side)
6. level 5 plus high density hydrographic and cultural features (urban areas)

The minimum size of the topographical data base required by this task is
1. 5 x 5 km or smaller
2. 10 x 10 km
3. 10 x 20 km
4. 10 x 30 km
5. 20 x 30 km
6. 30 x 30 km
7. 30 x 40 km

1

The environment in which the task is performed is
1. airport environment
2. desert, water, or other visually simple environment
3. countryside, forest, or visually similar environment
4. urban or other visually complex environment

2 3 4

Performance of the task requires
1. peripheral cues to provide "flow" information for judging and acceleration or for alerting the pilot
2. operation of the aircraft when severely pitched or rolled
3. identification of ground locations, such as landing fields, confined areas, targets, threats, and navigation waypoints/checkpoints
4. none of the above conditions

3

Ground locations that must be identified to perform this task are viewed
1. through the center 50% of the front window
2. through the center 75% if the front window
3. anywhere through the front window
4. through either the front or side windows

1

The content elements that would enhance the effectiveness of training on the task include
1. cultural lights
2. weapons blast
3. damaged vehicles
4. airborne vehicles
5. moving ground vehicles
6. smoke and dust
7. rotor wash
8. none of the above

2 3 4 5 6

Specific cues for performance of the task are provided by the following kinds of motion:

1. roll
2. pitch
3. yaw
4. vertical acceleration
5. lateral acceleration
6. longitudinal acceleration
7. there are no specific motion cues

7

The type of activity involved in the task is

1. continuous control movement
2. procedures
3. perception
4. decision making/rule using
5. using symbolic information
6. voice communication

1 2 3 4 5 6

The following audio signals (other than voice) provide cues for performance during the normal operation of this task:

1. correlated rotor flap
2. correlated engine noise (normal operation)
3. correlated engine noise (abnormal conditions such as engine failure, overspeed, and underspeed)
4. weapons firing
5. low/high RPM warning signal
6. skid noise (contact of skid with ground)
7. drive shaft/clutch failure
8. compressor stall
9. other audio signal
10. none of the above

4

Of the discrete audio signals (weapons firing, low RPM warning, skid noise, or clutch failure) that normally occur during the performance of this task, _______ are correlated with visual cues.

1. all
2. some
3. none

1

The task _____ the performance of an emergency procedure

1. Require
2. does not require

2

Please input the TASK NUMBER of the current task (four digit maximum)
: 1007

Performance of the task ____ an out-of-cockpit view

1. requires
2. does not require

1

When judging distances and ranges, the most convenient unit of measure is

1. feet
2. meters

2

Performing the task requires

1. absolute judgement of altitude in feet
2. absolute judgement of slant range in meters
3. judgement of clearance
4. detection of targets or navigation waypoints/checkpoints
5. slope landing
6. none of the above activities

3 4

The task requires judgement of

1. vertical clearance
2. lateral clearance
3. neither vertical nor lateral clearance

1 2

The task _____ the performance of an emergency procedure

1. requires
2. does not require

1

A discrete audio signal (weapon firing, low RPM warning, skid noise, or clutch failure) _____ for the initiation of the emergency procedure

1. provides a cue
2. does not provide a cue

2

APPENDIX G

INTERFACING ASTAR WITH A COST MODEL

Lucy C. Morse
Department of Industrial Engineering and Management Systems
University of Central Florida

ABSTRACT

This analysis develops a model for cost and training effectiveness of training systems. The proposed model considers the benefits resulting from the implementation of training devices and the traditional financial justification methods. Output from the Automated Simulator Test and Assessment Routine (ASTAR), which determines empirical measures of training effectiveness, is incorporated into the model. The structure of the model has the ability to consider a global approach which identifies and measures the desirability of the systems' strategic objectives as well as measuring the desirability of the tasks of a training device.

INTRODUCTION

This project is sponsored by the Army Research Institute (ARI) and the Army's Project Manager for Training Devices (PM-Trade) under a Broad Agency Announcement (BAA) Contract with the Institute for Simulation and Training.

The advance of military technology in recent years has resulted in an increased reliance on the use of training devices to achieve and maintain a state of combat readiness. In many cases, a training device is used to prepare personnel to operate a parent system which is too expensive, time consuming, or dangerous to be used for initial training. The success of training devices in reducng costs and training time for major weapon, aircraft, and other systems has led to the development of training devices for tasks that are not specific to a system. Training devices now are accepted and vital elements in the training programs of all branches of the service.

Although there is considerable amount of data about specific training devices, there is no organized body of information necessary to build cost effective and training effective systems. As a result the design of effective training devices is an effort that includes imperfect data, opinion-based design rules, and an increasingly large number of choices in the large array of technologies that can be used to address any single training problem.

One problem in the area of training is that research on methods to estimate training effectiveness and cost effectiveness lags behind the pressing need of the user to acquire and field effective systems. The goal of training device formulation is to develop a training device that meets the training requirements at minimum cost, or provides the maximum training benefit for a given cost. In the past a number of cost and training effectiveness studies have been completed.

Orlansky and String [5,6], Barcus [1], and Hofer [3] have concluded that flight simulators, computer-based instruction and maintenance training simulators appear to be as effective as the methods of training they can replace; also they reduce the costs of training. The trade-off between the training effectiveness of different components of a training device and the cost would need more complete data than was available at the time of their studies.

Understanding and correctly using cost estimating and cost and training effectivenss can help in the area of controlling the costs associated with training. It was concluded in the Barcus paper [1] that a through understanding of the costing process would help achieve the goal of attaining military readiness by providing the best training for the most reasonable cost.

Organized cost databases for training devices do not exist. Contractors are reluctant to share cost information because of the competitive arena in which they operate. Contract costs might be negotiated, or they might involve some new technological advances. Cost information is considered proprietary, also

engineering changes and modifications would affect the final cost.

The most promising model to study training effectiveness and cost effectiveness is "Optimization of Simulation - Based Training Systems" (OSBATS), which was developed and is described by Singer and Sticha [9]. The idea behind OSBATS is to organize the large amount of training technology and learning theory currently available and develop a model for aiding training developers in evaluating training device alternatives. This project is currently in the final prototype stage.

The OSBATS model is based on benefit and cost approximations that are used to analyze tradeoffs between various training device features in developing a device configuration, and then conducts similar tradeoffs between different training device configurations. The factors required by the OSBATS model to perform cost analyses include the device's investment cost, the fixed cost per year for operating the simulator, variable cost per hour for simulator use, life-cycle anticipated for the training device, and projected device utilization, in terms of hours per year. Additional factors are the instructional features acquisition cost, and the cost of minimum and maximum levels of fidelity in the visual systems and certain other simulator components.

Originally it was assumed that some of the cost data required by the OSBATS model had been collected by agencies within the Federal Government. Organized databases were not found to exist. Various estimating methods were used to provide

costs for the cost elements. [11] Partially due to lack of databases OSBATS is currently device specific, rather than generic.

Another model is the Automated Simulator Test and Assessment Routine (ASTAR), which is a computer-based decision aid that assists in evaluations of training device effectiveness. ASTAR converts information and judgments about various facets of a training system into forecasts of device effectiveness. An analyst provides a number of judgments or estimates in response to a variety of rating scales. These scales force consideration of different kinds of information about the training system or a part of the training system.

During a training system evaluation, four major ASTAR analyses are conducted. These are the training problem, acquistion efficiency, transfer problem, and transfer efficiency. The training problem defines the skill and knowledge deficiencies that the trainees would have relative to certain performance criterion. Acquisition efficiency examines the instructional features and training principles that help the trainees overcome their deficits with that particular training device.

The transfer problem determines the deficiency to performance criterion that remains after training on the device. Finally transfer efficiency indicates how well use of the training device will promote transfer of the learning that has occurred to the parent or actual equipment. [8]

These four areas can be conducted at three different levels

of analysis, ranging from macro to micro. If only general information is available Level 1 analysis or the macro analysis would be conducted. This score is the overall proportion of skills and knowledge to be learned. If a detailed analysis of skills and knowledge needed to learned of various components such as visual fidelity is desired, Level 2 would be used. The Level 3 score is the average deficit for each subtask.

At the end of the evaluation process, whether used at Level 1, 2, or 3, an analyst receives numerical estimates of device training effectiveness. These estimates are scores that are measures of the time and effort involved in attaining criterion performance on the training device. Cost effectiveness is not taken into consideration.

The objective of this study is to determine a rapid process to combine the scores from ASTAR and traditional financial justification methods to determine the training effectiveness and cost effectiveness of a training device or a specific part of a training device.

Attributes pertaining to training effectiveness would include learning difficulty, quality of acquisition, acquisition efficiency, residual deficit, residual learning difficulty, and transfer efficiency. The ASTAR results would be used in the next step. Attributes relating to cost factors would include the device's investment cost, fixed cost per year for operating the simulator, variable cost per hour for use, life-cycle, and projected utilization.

These attributes are then employed in a simple linear additive model, and a benefit/cost analysis is used with the results. The linear additive model is a decision tool that aggregates information from the independent attributes in a linear fashion to arrive at an overall score for each course of action being evaluated. In the past this model has been successfully used to justify automated manufacturing technologies. [7] The alternative with the lowest score is preferred in this case. The general form of the model is [10]:

$$V_j = \sum_{i=1}^{n} w_i x_{ij}$$

where V_j = the score for the jth alternative

w_i = the weight for ranking assigned to the decision attribute

x_{ij} = the ASTAR ranking.

To implement the methodology weights for ranking must be assigned to the attributes as shown in Table 1. This ranking expresses the relative importance of each attribute. These

weights are then combined with the results from ASTAR for attributes applying to training effectiveness, and to the cost factors.

Table 1. Ordinal scale weights for ranking the cost and training effectiveness attributes of the task under consideration.

Very important	1.00
Important	.75
Necessary	.50
Unimportant	.25

An application of ASTAR generates different types of ratings. Training Deficit, Training Difficulty, and Acquisition Efficiency ratings are combined to form an Acquisition score. The Transfer score is a combination of Residual Deficit, Residual Difficulty, Physical Similarity, Functional Similarity, and Transfer Efficiency. The Summary score is the sum of the Transfer score and the Acquisition score.

The Summary scores are measures of the time and effort involved in attaining criterion performance on the training device. The higher the score, the more time and effort will be needed to accomplish the training. For systems with the same training and operational objectives, the lower the Summary score, the "better" the device.

A training device is most effective if it reduces total training time to a minimum. The faster or more efficiently trainees reach and retain operational proficiency, the better the device. Using ASTAR, this measure pertains to the Transfer score and the Acquisition score; the lower the score, the better the device. There is no direct translation of the various ratings into hours of training. Comparisons between Transfer and Acquisition scores of different devices can only be made when the operational performance objectives are the same. For comparison purposes these scores are assumed to be ratios. The way to tell what the significant difference between two different scores on a single device is with a sensitivity analysis.

Phase I Methodology

In this phase a hierarchy of the simulation system's various device proposals may be structured, if there is a need to investigate various types of devices. Level 1 of ASTAR would be used.

Phase II Methodology

This phase evaluates various tasks of the simulation device selected in Phase I. For example, if the visual system of a training device is being examined, then the attributes will correspond to the initial cost, accuracy, training effectiveness, of several alternative configurations of that task and Level 2 of ASTAR would be used. The linear additive model used in Phase II combined with Benefit/Cost analysis is described as follows:

$$\text{Min}_j E_j = \frac{B_j = \sum_{i=1}^{p} U_i A_{ij}}{C_j = \sum_{i=1}^{q} U_i X_{ij}} \quad \text{for } j = 1 \text{ to } q$$

where

E_j = final score for each alternative

B_j = score representing the ability of the jth alternative to achieve the training objectives of the tasks

C_j = score representing the ability of the jth alternative to achieve the cost objectives of the specified tasks

U_i = an ordinal scale ranking assigned by the evaluator to the ith attribute to reflect its importance

A_{ij} = value assigned to the attribute by ASTAR

X_{ij} = traditional cost factor attributes

p = number of attributes common to all alternatives

q = number of alternative device configurations under consideration

The value in using ASTAR in combination with traditional financial justification methods is the rapid process which can be used to obtain training and cost effectiveness information about one or a few tasks of a training device. When one task is being examined only the questions in ASTAR which relate to that task need to be answered. Only the cost factors of that task must be determined.

Example

This training/cost effectiveness model could be used to examine the VIGS (Video Disk Interactive Gunnery Simulator) in more detail. The VIGS was built under contract for PM Trade by ECC in the mid-eighties. Its objective is to create a ballistically correct training device to teach a trainee the correct manual procedures to acquire and engage targets. Two important considerations with the current VIGS is that the targets are not interactive and the scenarios are canned, but may be preprogrammed.

If there were a need to improve a part of the visual system of VIGS, the methodology presented in this paper could be used to determine the "best" alternative in terms of training and cost effectiveness.

First, the tasks involved with the part of the visual system need to be identified and appropriate attributes selected. Once the attributes are selected and all changes in the training effectiveness for each alternative are specified, ASTAR can be run for each alternative. At the same time attributes for the cost factors for each of the alternatives would be calculated. Since only a partial task, in this example, is being determined, some of the cost factors, such as project utilization will be the same and can be ignored in this analysis. The attributes for the training effectiveness and cost effectiveness for the task are given in Table 2. The scale weights for ranking by one evaluator are also listed. (Data is hypothetical)

In the example the transfer efficiency for this visual task

Table 2. VIGS

i	Rank Ui	Attribute	Alternative 1 Ai1	UiAi1	Alternative 2 Ai2	UiAi2	Alternative 3 Ai3	UiAi3
1	1.0	Transfer Efficiency	1.84	1.84	1.06	1.06	4.6	4.6
2	.5	Acquistion Efficiency	17.16	8.58	19.28	9.64	17.31	8.655
		Bj =		10.42		10.7		13.255
3	1.	Investment Cost	10000	10000	8000	8000	12000	12000
4	.5	Fixed Costs	3000	1500	2500	1250	3000	1500
5	.2	Var. Costs	2000	400	3000	600	1000	200
		Cj =		11900		9850		13700
		Ej = Bj/Cj =		.000876		.001086		.0009675

is more important than the acquisition, and is ranked higher. The initial investment cost is considered to be very important and is given a weight of one. There is an increase in the service contract for this improvement and the service attribute must be considered as necessary. There is an unknown factor in the service contract and each contractor has estimated possible additional costs which are included with the variable cost attribute, but are considered basically unimportant in the scale weights.

The results of these variables and their weights from the ordinal ranking scale are shown on the spreadsheet given in Table 2. E_j is given for each alternative and the final decision is left to the evaluator. Once the initial analysis is completed, a sensitivity analysis may be performed by changing the ordinal scale weights for ranking.

CONCLUSION

This paper has presented a methodology to assist the decision maker in evaluating the training effectiveness and cost effectiveness using ASTAR. This may be used on an entire training device or a task of a training device or a subtask of that device.

In the future this study should be tested using actual data. Although cost data is hard to obtain, this method could be used quickly for a complete training device, since only a small part of the task needs to be examined. There is an opportunity for further research in this area.

BIBLIOGRAPHY

1. Barcus, George C. and Barcus, Theresa T. "Understanding Cost Estimating and Cost/Training Effectiveness Models - A Place to Start." Proceedings of the 8th Interservice/Industry Training Equipment Conference. Orlanddo, FL, November 1986.

2. Green, Eric, Dunlap, Scott, and Beger, Richard. "Methodologies for Comparing Cost/Effectiveness Measures of Alternative Training Systems". Proceedings of the 8th Interservice/Industry Training Equipment Conference. Orlando, FL, November 1986.

3. Hofer, Ronald, Kincaid, Peter, and Ozkaptan, Halim. "Training Systems R&D Program: Progress and Challenges". Proceedings of the 9th Interservice/Industry Training Equipment Conference. Washington, DC, November 1987.

4. Martin, M.F., Rose, A.M., and Wheaton, G.R. Applications for ASTAR in Training System Acquisitions. (P.O. Number C-44045). Washington, DC: American Institutes for Research. January 28, 1988.

5. Orlansky, J. and J. String. Cost-Effectiveness of Flight Simulators for Military Training, IDA Paper P-1275 Vol I and II, Institute for Defense Analyses, Alexandria, VA. August 1977 (AD A052 801).

6. Orlansky, Jesse, and String, Joseph. "The Cost-Effectiveness of Military Training." Proceedings of the 4th Interservice/Industry Training Equipment Conference. Orlando, FL, November 1982.

7. Parsaei, Hamid and Wilhelm, Mickey R. "A Justification Methodology for Automated Manufacturing Technologies." Presentation at Eleventh Annual Conference on Computers and Industrial Engineering, Orlando, March 1989.

8. Rose, A.M. and Martin, M.F. Implementation of ASTAR: Evaluation of the Combat Talon II Maintenance Trainer. (P.O. Number C-44045), Washington, DC: American Institutes for Research. February 16, 1988.

9. Singer, Michael and Sticha, Paul. "Designing Training Devices; The Optimization of Simulation-Based Training Systems". Proceedings of the 9th Interservice/Industry Training Equipment Conference. Washington, FL, November 1987.

10. Sullivan, W.G. "Models IEs Can Use to Include Strategic, Non-monetary Factors in Automation Decisions." Industrial Engineering, 18, 3, 169-176 (1986).

11. User's Manual for OSBATS. Spring, 1989.

0000050