

DTIC FILE COPY
Technical Report 904

1

AD-A227 361

Research and Development on the Characterization of Simulation-Based Training Systems: Project Executive Summary

Paul J. Sticha
Human Resources Research Organization

July 1990

DTIC
ELECTE
OCT 02 1990
S B D



**United States Army Research Institute
for the Behavioral and Social Sciences**

Approved for public release; distribution is unlimited

U.S. ARMY RESEARCH INSTITUTE FOR THE BEHAVIORAL AND SOCIAL SCIENCES

**A Field Operating Agency Under the Jurisdiction
of the Deputy Chief of Staff for Personnel**

EDGAR M. JOHNSON
Technical Director

JON W. BLADES
COL, IN
Commanding

Research accomplished under contract for
the Department of the Army

Human Resources Research Organization

Technical review by

John A. Boldovici
Ray S. Perez

NOTICES

DISTRIBUTION: Primary distribution of this report has been made by ARI. Please address correspondence concerning distribution of reports to: U.S. Army Research Institute for the Behavioral and Social Sciences, ATTN: PERI-POX, 5001 Eisenhower Ave., Alexandria, Virginia 22313-5600.

FINAL DISPOSITION: This report may be destroyed when it is no longer needed. Please do not return it to the U.S. Army Research Institute for the Behavioral and Social Sciences.

NOTE: The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
1a. REPORT SECURITY CLASSIFICATION Unclassified			1b. RESTRICTIVE MARKINGS ---		
2a. SECURITY CLASSIFICATION AUTHORITY ---			3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.		
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE ---					
4. PERFORMING ORGANIZATION REPORT NUMBER(S) FR-PRD-88-28			5. MONITORING ORGANIZATION REPORT NUMBER(S) ARI Technical Report 904		
6a. NAME OF PERFORMING ORGANIZATION Human Resources Research Organization		6b. OFFICE SYMBOL (If applicable) HumRRO	7a. NAME OF MONITORING ORGANIZATION U.S. Army Research Institute for the Behavioral and Social Sciences		
6c. ADDRESS (City, State, and ZIP Code) 1100 South Washington Street Alexandria, VA 22314-4490		7b. ADDRESS (City, State, and ZIP Code) U.S. Naval Training System Center 12350 Research Parkway Orlando, Florida 32826-2376			
8a. NAME OF FUNDING/SPONSORING ORGANIZATION U.S. Army Research Institute for the Behavioral and Social Sciences		8b. OFFICE SYMBOL (If applicable) PERI-I	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER MDA903-85-C-0169		
8c. ADDRESS (City, State, and ZIP Code) 5001 Eisenhower Avenue Alexandria, VA 22333-5600		10. SOURCE OF FUNDING NUMBERS			
		PROGRAM ELEMENT NO. 63744A	PROJECT NO. 795	TASK (3104) NO. 345	WORK UNIT ACCESSION NO. C1
11. TITLE (Include Security Classification) Research and Development on the Characterization of Simulation-Based Training Systems Project Executive Summary					
12. PERSONAL AUTHOR(S) Sticha, Paul J.					
13a. TYPE OF REPORT Final		13b. TIME COVERED FROM 85/04 TO 88/10		14. DATE OF REPORT (Year, Month, Day) 1990, July	
15. PAGE COUNT					
16. SUPPLEMENTARY NOTATION Stephen L. Goldberg, contracting officer's representative					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB-GROUP	Training system optimization Fidelity		
			Training system model Instructional features		
			Medium selection Training resource allocation		
19. ABSTRACT (Continue on reverse if necessary and identify by block number) For this project, researchers developed a model for the optimization of simulation-based training systems (OSBATS) using a systematic, top-down design procedure. The model consists of five tools that address the following problems: (a) determining which tasks should be trained by part-mission or full-mission simulators and which should be trained on actual equipment; (b) specifying which instructional features are needed to train a set of tasks efficiently within a budgetary constraint; (c) specifying the optimal levels at which fidelity should be provided along several fidelity dimensions to meet training requirements and satisfy a training-device cost limit; (d) determining the group of training devices that can train all required tasks at the minimum cost; and (e) determining the optimal allocation of training time to training devices, given constraints on device use. The tools share a common database of task requirement, training device, and cost data. A prototype decision					
(Continued)					
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION Unclassified		
22a. NAME OF RESPONSIBLE INDIVIDUAL S. Goldberg			22b. TELEPHONE (Include Area Code) (407) 380-4690		22c. OFFICE SYMBOL PERI-IF

ARI Technical Report 904

19. ABSTRACT (Continued)

support system (DSS) implementing the OSBATS model was developed and formative evaluation of the model and software conducted. The model was applied to a problem in Army aviation, and specifications for its application to armor maintenance were developed.

The OSBATS model is a set of tools that can help the engineer perform the tradeoff analyses needed to support the selection of the best technical approach to a training-device design. This report gives an overview of the model using the IDEFO (Integrated Computer-Aided Manufacturing Definition) system modeling language. The IDEFO model provides a top-down analysis of major model components and their relationships. The OSBATS model has been implemented as a decision support system (DSS) that runs on an IBM PC/ACT, Zenith 248, or compatible computer. The report describes the results of a formative evaluation and an analysis of the activities required to apply the OSBATS model to the armor maintenance domain.

The OSBATS model may be used by an engineer responsible for the development of a training-device concept to perform tradeoff analyses required to support the selection of the best technical approach to the training-device design. The prototype DSS provides an interactive environment in which the engineer may perform several kinds of tradeoff analyses. The OSBATS software includes the data necessary to use the model for certain problems in Army rotary-wing aviation. The model processes will generalize readily to other training domains when the required data have been obtained.

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

Technical Report 904

**Research and Development on the Characterization of
Simulation-Based Training Systems:
Project Executive Summary**

Paul J. Sticha

Human Resources Research Organization

**PM Trade Field Unit at Orlando, Florida
Stephen Goldberg, Chief**

**Training Research Laboratory
Jack H. Hiller, Director**

U.S. Army Research Institute for the Behavioral and Social Sciences
5001 Eisenhower Avenue, Alexandria, Virginia 22333-5600

Office, Deputy Chief of Staff for Personnel
Department of the Army

July 1990

**Army Project Number
2Q263744A795**

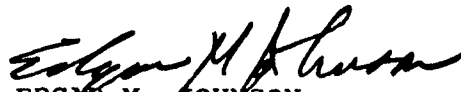
Training Simulation

Approved for public release; distribution is unlimited.

FOREWORD

The cost of training devices and simulators has exceeded, in some cases, the cost of the operational equipment that they service. The capabilities for simulating reality are simultaneously increasing on an annual basis. The problem confronted by the military is to determine exactly how much simulation is sufficient for the stated learning objectives. Behavioral and analytical techniques that can quickly project or predict how much simulation and training is required are lacking. At the same time information on the cost-effective use of training equipment within courses of instruction is sparse. The development of models, databases, and techniques addressing these problems provides the first steps toward providing integrated behavioral and engineering decisions in designing, fielding, and using advanced training technology. The potential effect on the Army is to reduce the cost of training equipment while increasing the equipment's instructional effectiveness.

In response to these concerns and problems, the Army Research Institute for the Behavioral and Social Sciences (ARI) and the Project Manager for Training Devices (PM TRADE) have joined efforts. PM TRADE has maintained partnership in all aspects of the development of the models, databases, and analytical techniques. The final prototype software was delivered to ARI and PM TRADE in December 1988, and has been disseminated to interested parties at Fort Rucker, the Army Training Support Command, and the Systems Training Directorate at the Training and Doctrine Command. The prototype has also been provided at their request to the Naval Training Systems Center Human Factors Research Group, the Air Force Aeronautical Systems Division, the Air Force Human Research Laboratory at Williams AFB, and National Aeronautics and Space Administration Ames Research Center. The models and techniques developed in this effort are expected to provide the basis for useful aids supporting the integration of behavioral and engineering data, knowledge and expertise in training equipment design in the future.


EDGAR M. JOHNSON
Technical Director

ACKNOWLEDGMENTS

The author would like to acknowledge the support received in completing this project and preparing the reports. Dr. Halim Ozkaptan, the technical monitor for the project, provided sage counsel and guidance that set the direction for this effort. Dr. Michael J. Singer monitored the ongoing activities of the project, and his technical contributions have been valuable. Mr. William Goodrick provided an ongoing link between the model developers and the user community at PM TRADE. Other engineers from PM TRADE evaluated the OSBATS software at various stages in its development. Many of their suggestions have been incorporated into the software.

Many members of the contractor research staff have contributed to this effort. Mr. H. Ric Blacksten and Dr. Dennis M. Buede were chiefly responsible for the development and documentation of individual OSBATS modules. Dr. C. Mazie Knerr provided management support at several critical points in the project and also worked with Dr. John E. Morrison on the development of the instructional-feature selection rules. Ms. Elizabeth L. Gilligan was responsible for model documentation, model verification, sensitivity analyses, and the preparation of the User Guide. She was assisted in these efforts by Mr. B. Leon Elder and Mr. Peter W. DeYoe. Dr. Randall J. Mumaw was responsible for the formative evaluation effort. Ms. Marty Carson was responsible for document processing and preparation.

We are pleased with the working relationship we were able to develop and maintain with our subcontractors. The quality of the prototype software is due primarily to the efforts of Mr. James Stock from PAR Government Systems Corporation. Dr. Kenneth Cross and Mr. Carl Bierbaum provided the knowledge that was incorporated into the fidelity rule base. They were also responsible for developing the data used in the example problem.

RESEARCH AND DEVELOPMENT ON THE CHARACTERIZATION OF SIMULATION-BASED TRAINING SYSTEMS: PROJECT EXECUTIVE SUMMARY

EXECUTIVE SUMMARY

Requirement:

The goal of this project was to develop and demonstrate methods for helping the training-device designer perform the tradeoff analyses required for training-device design. These methods should allow the designer to determine the training-device alternatives that meet training requirements at a minimum cost or provide the maximum training effectiveness at a given cost. The methods should apply to the concept-formulation phase of the training-device development process and should be usable by the engineer responsible for developing the training-device concept.

Procedure:

This report briefly introduces the model development, a formative evaluation, and the potential application of the Optimization of Simulation-Based Training Systems (OSBATS) model to the armor maintenance domain. The model for the OSBATS was developed using a systematic, top-down design procedure. An overview of the model was developed using the Integrated Computer-Aided Manufacturing Definition (IDEFO) system modeling language. The IDEFO model provides a top-down analysis of major model components and their relationships. A prototype decision support system (DSS) implementing the OSBATS model was developed and formative evaluation of the model was conducted using the software. The model was demonstrated on a problem in Army aviation, and specifications for its application to armor maintenance were developed.

Findings:

The OSBATS model is a set of tools that can help the engineer perform the tradeoff analyses needed to support the selection of the best technical approach to a training-device design. The model consists of five tools that address the following problems: (a) determining which tasks should be trained by part-mission or full-mission simulators, and which should be trained on actual equipment; (b) specifying which instructional features are needed to train a set of tasks efficiently within a budgetary constraint; (c) specifying the optimal levels at which fidelity should be provided along several fidelity dimensions in order to

meet training requirements and satisfy a cost limit; (d) determining the most effective group of training devices for training tasks at the minimum cost; and (e) determining the optimal allocation of training time to training devices, given constraints on device use. The tools share a common data base of task requirements, training device information, and cost data. The results of the formative evaluation led to corrections in the model as developed and implemented. The analysis of armor training indicated that the model could be applied, given complete data and rules for the new domain.

Utilization of Findings:

The OSBATS model has been implemented as a prototype DSS that runs on an IBM PC/AT, Zenith 248, or compatible computer. The OSBATS model potentially may be used by anyone responsible for the development of a training-device concept to perform tradeoff analyses required to support the selection of the best technical approach to the training-device design. The prototype DSS demonstrates an interactive environment in which an engineer may perform several kinds of tradeoff analyses. The OSBATS software includes the data necessary to use the model for certain problems in Army rotary-wing aviation. The model processes should generalize readily to other training domains, when the required data and rules have been obtained.

RESEARCH AND DEVELOPMENT ON THE CHARACTERIZATION OF SIMULATION-
BASED TRAINING SYSTEMS: PROJECT EXECUTIVE SUMMARY

CONTENTS

	Page
INTRODUCTION	1
Background.	1
The Training-System Design Problem.	2
OSBATS Model Overview	5
OSBATS Implementation	9
Organization of This Report	10
OSBATS MODEL DESCRIPTION	11
User Perspective.	13
Summary of Data Requirements.	15
System Modeling Methods	20
OSBATS SYSTEM MODEL.	28
OSBATS MODEL IMPLEMENTATION.	59
General Features of the Software.	60
Summary of Module Displays.	61
Summary of User Functions	64
EVALUATION OF THE OSBATS MODEL	67
Structured Interviews of Potential Users.	68
Sensitivity Analyses.	70
Tank Turret Mechanics Analysis.	72
SUMMARY AND CONCLUSIONS.	75
Significant Accomplishments of This Research.	75
Needs for Future Model Development.	77
REFERENCES	81

LIST OF TABLES

Table 1. IDEFO node list.	29
2. Matrix representation showing incremental benefit-to-cost ratios for fidelity levels	63

CONTENTS (Continued)

	Page
Table 3. Package representation showing optimal training-device design at a cost of \$5.5 million	64

LIST OF FIGURES

Figure 1. Interaction of training system components	4
2. Example of a hierarchical, top-down model	22
3. IDEFO node numbering convention.	23
4. Sample IDEFO diagram.	25
5. Sample IDEFO diagram showing box and arrow syntax.	26
6. Sample IDEFO diagram showing source and destination	27
7. Graphical representation of fidelity optimization model results.	63

RESEARCH AND DEVELOPMENT ON THE CHARACTERIZATION OF SIMULATION-BASED TRAINING SYSTEMS: PROJECT EXECUTIVE SUMMARY

Introduction

The increasing cost of training and limitations in the military training budget have led to increased emphasis on training cost-effectiveness. In addition, advances in instructional technology have greatly increased the options that are available to the training-system designer. Current training system design processes do not address the cost-effectiveness of the wide range of training-device and simulator options available to the training designer. This report describes a system of models for the optimization of simulation-based training systems (OSBATS). The OSBATS model provides a structure for the decision-making processes involved in training-system design. The recommendations of the model are based upon the effectiveness, efficiency, and costs involved in training-device development and use. The OSBATS system provides a coherent set of procedures for decision making and a set of tools to aid the designer in following these procedures.

Background

The U.S. military invests a considerable amount of resources for training, both by training institutions and in operational units. This training provides soldiers the skills required to operate and maintain complex modern weapon systems. According to the Military Manpower Training Report for Fiscal Year 1988 (Office of the Assistant Secretary of Defense, Force Management and Personnel, 1987), the cost of military training conducted by training institutions for fiscal year 1988 is estimated to be more than \$18 billion. This figure includes \$7.1 billion for training areas related to weapon-system operation and maintenance. Analyses of the total military budget indicate that the magnitude of unit training is at least as great as that of institutional training (DoD Training Data and Analysis Center, 1985). Thus, the total annual cost for institutional and unit training probably exceeds \$34 billion, with perhaps \$14 billion of this training directly related to the operation and maintenance of weapon systems. Given the magnitude of military training, the importance of cost-effective training is clear. An improvement in training efficiency as small as 1% could save \$340 million annually.

Many of the reasons for the high cost of military training are obvious. Weapon systems required for hands-on training are expensive to procure and operate. Other required equipment, such as ammunition, is also expensive. In addition, training of many tasks requires special conditions that replicate the battle environment, equipment malfunctions, opposing force activities, and special environmental situations that provide critical cues

for weapon system operation and maintenance. Associated with the cost of producing these special training conditions are limitations on the availability of training ranges, ammunition, and so forth, as well as safety considerations.

Advances in instructional technology, such as computer-generated imagery, computer-assisted instruction, interactive videodisc, and simulation technology have made simulation-based training possible for a wider range of skills. These advancements have increased the number of options available to the training designer. The overall effect of the increased number of training options has been to make the design task more difficult. The designer must consider different training strategies (that is, a part-task training strategy, a full-mission simulator, or actual equipment training possibly enhanced with embedded training), more or less sophisticated training-device designs, and specific allocation of training times to training devices. The training-system designer needs to have a formal training-system design process and tools to aid in the performance of this process.

The Training-System Design Problem

The goal of the OSBATS model is to provide tools that allow the training-system designer to produce cost-effective training-system designs. The definition of the term "training system" that we have adopted has had a great impact on the types of procedures that we have incorporated into the OSBATS model. In addition, we have made some restrictions of the types of issues that will be addressed by the model. The following subsections provide the basic problem definition we have adopted.

Definition of a Training System

Definitions of what constitutes a "training system" vary from very broad to quite specific. So that we may make some progress in reaching a solution that optimizes training-system design, we will need to be somewhat limited in our definition of a training system. We realize that when we make this definition, the training system that is the concern of the OSBATS model is really a subsystem of a larger system.

We define a training system as a set of activities designed to give students the skills needed to operate or maintain a weapon system. From this definition, we may distinguish the following system components.

1. Target weapon system or job. We are primarily concerned with training for the operation and maintenance of weapon systems, because this is where the potential for the use of training devices is the greatest.

2. Training requirements. The training requirements are the duties or tasks that must be performed to set standards at the conclusion of training.
3. Student population. The students being trained are characterized by to their knowledge and skills. We anticipate that different kinds of training may be appropriate for initial skill training, transition training, continuation training, functional training, unit training, and so forth.
4. Trainer. The trainer includes both the instructors who deliver the training and the organizational entity responsible for training development.
5. Training methods and devices. Training methods define training strategies and the mix of training media. Training devices are characterized by the extent to which they represent elements of the actual equipment or job environment and by the instructional support features they possess.

Figure 1 illustrates how these components interact to define the training system. The first two components define the controls on the training system considered in the definition. By restricting our attention to training for a single target weapon system, we can deal with single training courses. We are not concerned with problems of allocating training to settings, or with a soldier's career progression through several MOS, although both of those problems have a critical impact on the overall cost and effectiveness of training. The training requirements control the training system by specifying the criteria for successful operation of the training system.

The third component defines the inputs to the training system. The student population characteristics define the extent of the training problem, by specifying the skills of the students who enter training. The scope of the training problem is determined by the difference between the students' entering skills and the required skills following training.

The final two components represent the mechanisms by which the training is accomplished. Of these components, only the training methods and devices include variables over which we have control in the design of a training system. Those variables that are related to training-device design and use are the concern of the OSBATS model. In general, these variables include the fidelity of training devices, the instructional features incorporated in them, and the assignment of training time to training devices.

We judge the optimality of a training system by the cost required to meet the training requirements. The major concern of the OSBATS model is the design and use of training equipment, particularly equipment that simulates the operation of part or all of the weapon system. In general, we want to minimize the training cost required to meet the training requirements. We may also be concerned with obtaining the maximum training effectiveness for a specified cost.

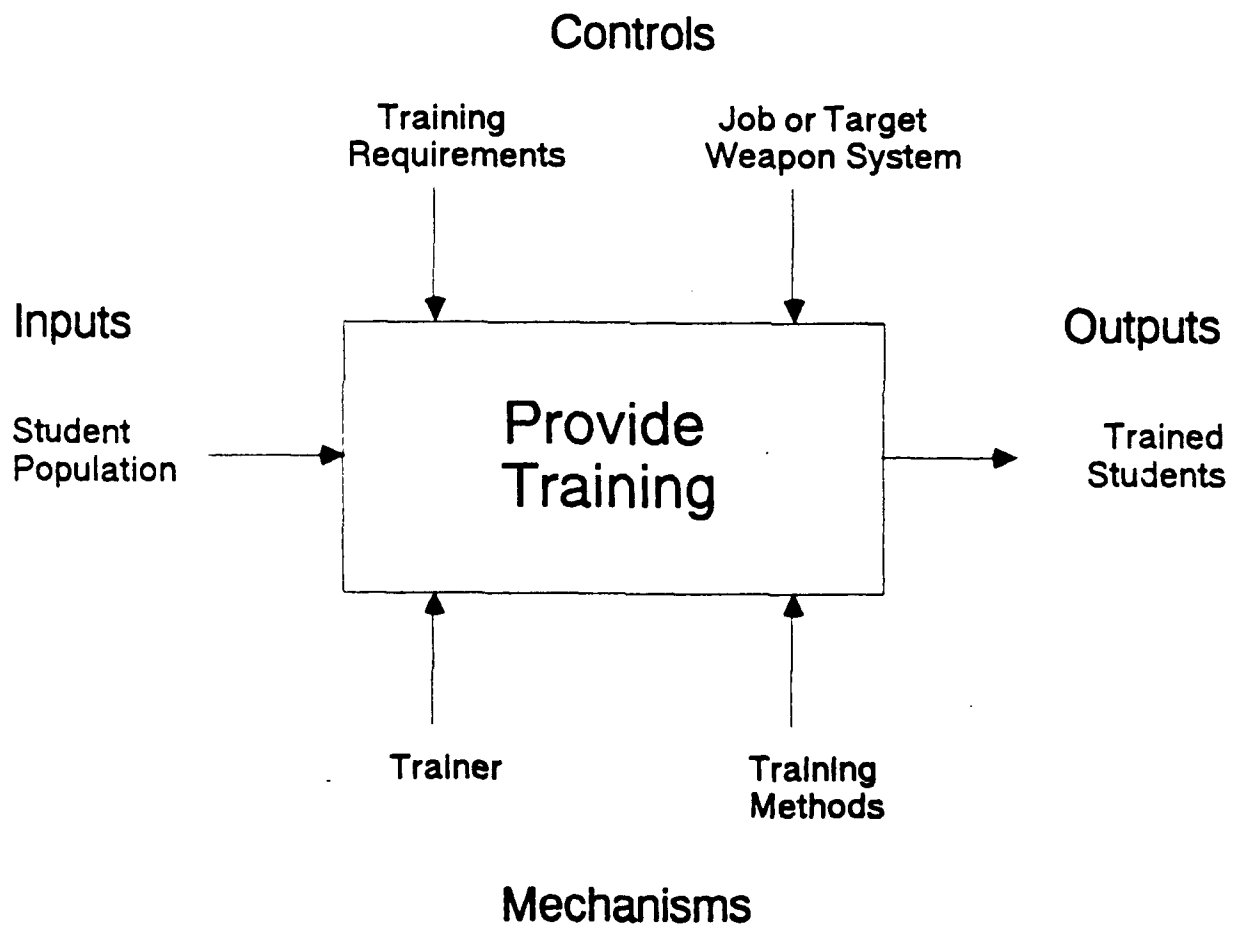


Figure 1. Interaction of training system components.

Scope of the OSBATS Model

Within the general framework described above, we have restricted the scope of the OSBATS model in several respects.

1. OSBATS focuses on tasks that can be trained by training devices or simulators. The models do not address classroom training issues.
2. OSBATS is concerned with training devices that interact with the student dynamically in a manner that is analogous to the interactions that occur with actual equipment. Training media as movies, videotapes, static representations of actual equipment, and other training aids that serve primarily to enhance classroom training are not addressed. OSBATS is concerned with computer-based training (CBT) to the extent that the training involves a dynamic representation of the tasks being trained, rather than a static presentation of information.
3. The OSBATS models address institutional training issues rather than unit, team, or collective training. Unit training involves complexities that were judged to be too difficult to handle in the initial development of the models. However, it should be possible to generalize the model procedures to apply to unit training at a later date.

OSBATS Model Overview

The goal of the OSBATS model is to provide methods to produce training-device designs that meet the training requirements at the minimum cost. The proposed user of this model is the system engineer responsible for the formulation of a training-device design concept. The OSBATS model provides tools to aid the tradeoff analyses required to support the selection of a best technical approach to a training-device design. Using the OSBATS model, the user can perform comparative analyses that identify cost drivers, produce and evaluate alternative training-device design concepts, and specify cost-efficient ways to use training devices to meet the training requirements.

Five modules interact to help the engineer develop and evaluate training-device concepts. The engineer can use the modules singly or in combination to address a wide variety of training-device design issues. The following list describes some of the analyses that can be performed using the OSBATS model.

1. Screen training requirements to determine which requirements can be met most appropriately using some kind of training device.
2. Identify tasks that can be trained adequately using a simple, inexpensive training device.

3. Compare the thousands of potential training-device design options to determine which ones meet the specific task training requirements at the lowest cost.
4. Examine the minimum fidelity levels required to train a task, based on the specific activities performed as a part of the task.
5. Determine which instructional support features are needed to maximize the efficiency with which the training requirements may be met on a specific training device.
6. Compare the cost effectiveness of training conducted using a sophisticated, full-mission simulator with training conducted using a combination of simpler, part-mission training devices.
7. Compare the cost-effectiveness of a design proposed by the user or other individual with a design of the same cost recommended by the model.
8. Determine how training time should be allocated among training devices and actual equipment.
9. Investigate the effect of limited availability of actual equipment or a training device on the training time and cost required to meet the training requirements.

Overall Modeling Framework

The OSBATS model incorporates several modeling techniques to aid the training-system designer. The overall modeling framework is based on methods that define the training strategy that meets the training requirements at the minimum cost. This framework was originally described by Roscoe (1971) and has been extended by Povenmire and Roscoe (1973), Carter and Trollip (1980), Bickley (1980), Cronholm (1985), and our own work (Sticha, Blacksten, Buede, & Cross, 1986; Sticha, Singer, Blacksten, Mumaw, & Buede, 1987). In its simplest form, the method compares the ratio of effectiveness of two training alternatives to the ratio of cost of the options. For example, if a training program that employs one hour of training on a simulator saves 30 minutes of training on actual equipment, and the hour of simulator training costs as much as 20 minutes of training on actual equipment, then the simulator will meet the training requirement at a lower cost than actual equipment. Thus, the approach addresses the tradeoff between the increased training time that is usually required to use a simulator and the decreased cost of that time.

We have extended the basic modeling framework in two ways to produce two model tools that address the selection of training devices from multiple candidates for multiple tasks, and the allocation of training time to the selected devices. Both extensions make the same assumptions about learning and transfer processes. The first extension makes simplifying assumptions about training cost so that it can provide an interactive environment for addressing training-device selection alternatives. The second extension relaxes some of the assumptions to allocate training resources to training devices considering both discrete purchase costs and device use constraints.

Task Clustering Model

The general resource allocation modules are supplemented by three other tools. The first of these tools reviews task requirements, simulation needs, and cost of simulation capability in order to define clusters of tasks that have similar simulation requirements. The method currently defines the following three classes of training devices: (a) a full-mission simulator (FMS) that simulates many or all of the subsystems of the actual equipment, (b) one or more part-mission simulators (PMSs) that simulate selected equipment subsystems, or (c) actual equipment.

This evaluation examines device-unique capabilities, such as training in unsafe situations, and cost savings to establish the value of training with some sort of training device. In addition, the task requirements for fidelity are used to estimate the development cost that would be required to achieve the required fidelity for each task. Using the assessed costs and benefits, the model sorts the tasks into three clusters: (a) those tasks that should be trained on actual equipment because the benefits of simulation do not justify the expense required to develop an effective training device, (b) those tasks for which training in a simulated environment is cost-effective and which have limited cue and response requirements so that they require only a PMS, and (c) those tasks for which training in a simulated environment is cost-effective, and which require an FMS because they require a high-fidelity representation of the environment on several dimensions.

The tool makes its major recommendation by comparing the required development cost of the training device to the potential operating-cost savings brought about by its use. If the operating-cost savings is sufficient to recover the development cost over the life cycle of the weapon system, the model will recommend the use of a training device. Otherwise, the model will recommend that actual equipment be used. The recommendations of the economic analysis are overridden, however, if a training device is required for safety considerations.

Training-Device Design Models

The task clusters defined by the above procedure provide the requirements used to design individual training devices. The task at this point is to develop training device designs that have the fidelity and instructional features required to meet the training requirements for the tasks in a single cluster while avoiding extraneous or inefficient features. We have applied a general design methodology to the analysis for training-device design. This methodology addresses problems in which there are a large number of alternatives formed by the factorial combination of several dimensions. We have developed two applications of this methodology. The first application addresses the instructional features that should be included in the training device; the second application addresses the fidelity features that should be included.

The model views instructional features as elements of training devices that can improve training efficiency on individual tasks. That is, instructional features reduce the time or cost required to achieve a given performance level on a training device. They do not affect the ultimate level of actual-equipment performance that can be reached by using a training device. The number of tasks aided by each instructional feature forms the basis of an index of benefit for the feature. The analysis proceeds by comparing the benefit to the cost of incorporating each instructional feature into the training device. The analysis then orders the features according to the ratio of benefit to cost. This order specifies the optimal collection of instructional features as a function of the total budget for instructional features. The appropriate budget for instructional features, given a total training-device budget, is determined in the following model.

The same modeling framework is then used to address how much should be invested in the fidelity of the training device being designed. The model considers several dimensions of fidelity that describe task cue and response requirements. The task requirements on the fidelity dimensions are compared to the cost of meeting these requirements to determine the dimensions for which increased fidelity is justified by increased training effectiveness. The output of this model is a set of possible training-device configurations applicable to the task set, each of which offers the greatest effectiveness for its cost.

The model makes its selection based on the incremental benefit/cost ratio of the fidelity dimension levels. The costs are calculated from the fidelity levels, and represent development costs. The benefits are calculated from the number of tasks for which each level of the fidelity dimensions would be adequate, based on the technical performance associated with each option and the cue and response requirements of the tasks from the fidelity dimensions.

OSBATS Implementation

A major component of this project is the development and revision of software implementing the OSBATS model. The OSBATS software is a prototype decision support system (DSS) that interacts with the training-system designer. The system provides an interactive environment in which the training-system designer may examine and evaluate alternative training-system designs. The software contains a data base that describes the tasks, fidelity dimensions and levels, instructional features, and cost factors that relate to the AH-1 Airman Qualification Course (AQC).

The software implements all of the modeling capabilities represented in the model design. The software allows the user to define task clusters, develop candidate training-device designs for individual task clusters, and evaluate the cost of providing training using these designs in various combination with or without actual equipment. The data base management requirements for the OSBATS have been investigated and a prototype developed in another contract (Willis, 1988).

The OSBATS software runs on a Zenith 248, IBM PC/AT or compatible with 640K of memory and a 10 megabyte hard disk. In addition, the following features are required: (a) an enhanced graphics adapter (EGA) and color EGA monitor, (b) an 80287 numeric coprocessor, and (c) a Microsoft-compatible mouse.

OSBATS Documentation

This report summarizes all general aspects of the development, implementation, and evaluation of the OSBATS model. In addition to this summary, we have written the following four reports that describe specific aspects of the OSBATS model in greater detail.

Sticha, P.J., Blacksten, H.R., Buede, D.M., Singer, M.J., Gilligan, E.L., Mumaw, R.J., and Morrison, J.E. Optimization of Simulation-Based Training Systems: Model Description, Implementation, and Evaluation (Final Report FR-PRD-88-26).

Sticha, P.J., Singer, M.J., Blacksten, H.R., Morrison, J.E., and Cross, K.D. Research and Methods for Simulation Design: State of the Art (Final Report FR-PRD-88-27).

Gilligan, E.L., Elder, B.L., and Sticha, P.J. Optimization of Simulation-Based Training Systems: User's Guide (Research Product RP-PRD-88-30).

Stock, J.A., Gilligan, E.L., and Sticha, P.J. Optimization of Simulation-Based Training Systems: Programmer's Guide (Research Product RP-PRD-88-29).

Organization of This Report

The remainder of this report summarizes all aspects of the project. The next section describes the methods that were incorporated into the OSBATS model. Following that section, we describe the specific capabilities of the computer implementation of the model. The following section describes the results of the formative evaluation activities that were carried out. The final section summarizes the accomplishments of this effort and describes the future for the OSBATS model, including both the requirements for technology transfer and the needs for research.

OSBATS Model Description

The OSBATS model was developed iteratively using a top-down, system-analytic approach. We decomposed the overall goal of optimizing training-device based training into three subgoals, and developed a set of tools to meet these subgoals.

The first subgoal is to identify tasks that are good candidates for training using a training device. Tasks may be candidates for device-based training for several reasons. First, the use of a training device may provide training at a lower cost than comparable training on actual equipment. Second, a training device may be able to produce special environmental conditions that would be unsafe, expensive, or impossible to produce using actual equipment. Finally, a training device may be more efficient by allowing the student more repetitions of the tasks during training than actual equipment, or by using appropriate instructional features. A second element of this subgoal is to determine clusters of tasks that have similar training-device needs. The task clusters produced by this process form the requirements used as the basis for training-device design.

The second subgoal is to specify the functional characteristics of training devices with a level of sophistication and cost that is tailored to the requirements of the tasks for which they are designed. The major training device components considered in this problem either simulate the equipment and environment or provide instructional support to the training process. The simulation components may vary with respect to the fidelity with which they represent corresponding actual-equipment components. There are many simulation components, such as the device's visual system or motion system, to be considered in generating a training-device configuration. The value of investing in different levels of fidelity for these components depends on the effectiveness of the components in reaching the training requirements as well as the cost of the components. The device-design process must first determine the minimum level of fidelity required by the tasks to be trained. Then the cost of the fidelity levels is considered in order to select the most cost-effective set of components for the configuration. The effectiveness of instructional features depends upon the characteristics of the tasks to be trained and the population of students. As with fidelity, the training device should be designed with instructional features that provide the greatest improvement in training for the least cost.

The third subgoal is to determine the way to allocate training resources among existing and proposed training devices, and actual equipment that minimizes training cost. In some situations, it may be possible for a training device to provide cost-effective training on tasks other than those for which it was designed. In this case, the training device should be used for those tasks. In other situations, it may not be possible to

provide the required fidelity to train a task at an acceptable cost. In this case, it may be optimal to design a simpler training-device that would replace only a portion of the training time on actual equipment. In its complete formulation, a procedure for allocating training among training devices and actual equipment must consider the constraints on the use of training devices and actual equipment that come from budgetary limitations, space and equipment availability, and safety concerns.

Some of the complexity of training-system design is caused by interactions between the three subgoals. That is, although there is a general logical progression through the subgoals, later processes can provide feedback to earlier processes. For example, the resource allocation process that addresses the third subgoal may indicate that a high-cost simulator design leads to a lower overall training cost than either a moderate- or a low-cost device. This feedback may lead the analyst to develop and evaluate other high-cost device designs. On the other hand, the resource allocation process might indicate that a low-cost training device can provide adequate training effectiveness for all but a small subset of the tasks. This result might prompt the analyst to design a new training-device specifically tailored to the tasks that could not be trained by the low-cost device.

The interactions between the subgoals for the OSBATS model imply that a simple linear approach to the problem will not work in some cases. Because of the complexity of the subgoal interactions, the OSBATS model must be designed to be used iteratively. That is, the results of individual model components must provide input to later components and feedback to earlier components. The OSBATS model provides for iterative application of its component modules with greater precision at each application cycle. The subgoal interactions also indicate the need for sensitivity analyses in which model assumptions are varied to ensure that the solution obtained is a global, rather than local, optimum.

We developed five software tools to address the three subgoals. One tool, the Simulation Configuration Module, addresses the first subgoal. Two tools, the Instructional Feature Selection and Fidelity Optimization Modules, address the second subgoal. Two tools, the Training Device Selection and Resource Allocation Modules, address the third subgoal. The function of each tool is briefly described below.

1. The Simulation Configuration Module clusters tasks to be trained according to their need for training on a full-mission simulator (FMS), one or more part-mission simulators (PMSs), or actual equipment (AE).
2. The Instructional Feature Selection Module determines the relative priority with which instructional features should be included in a training device.

3. The Fidelity Optimization Module determines the relative priority of features that allow a training device to represent the actual equipment and operational environment.
4. The Training Device Selection Module selects a set of training devices that can be used to meet the training requirements for each task at the least cost.
5. The Resource Allocation Module determines the optimal allocation of training time to training devices and actual equipment to meet all training requirements, considering constraints on device procurement and use.

We continued to use the top-down structuring methods to develop each tool. First, we developed general procedures and analysis strategies. The general procedures specify the kinds of variables that are relevant to the tool and how they are combined. For example, the general procedures specify whether a tool considers life cycle cost or development cost only, what factors are considered in the determination of effectiveness, and whether the recommendation of the tool is based on an effectiveness/cost ratio or another mathematical optimization procedure. Then we formulated specific procedures by examining the current knowledge and supplementing this knowledge, where necessary, with reasonable conjectures. The specific procedures provide the detailed methods used to calculate relevant cost and effectiveness measures. The resulting tools are general in that they should apply to a wide variety of training systems, simulators, and other training devices. However, the data used by the model include information that is specific to our initial application in advanced Rotary-Wing Aviation training.

User Perspective

The concept of operation for the OSBATS model is based on the iterative use of the five model tools to make recommendations regarding the definition of task clusters, the design of training devices, and the allocation of training resources among selected training devices. Both the subset of tools that are used and the order in which they are used may depend on the requirements of the problem and the preferences of the user. Although the user may apply the tools in a variety of orders, the most natural order is the order in which the tools were listed above. The following text describes an application of the tools in that order.

The analyst would use the Simulation Configuration Module first to provide a preliminary recommendation for the use of either actual equipment or one or more training devices, based on the training requirements. This analysis would produce three clusters of tasks. Two of these clusters define tasks for which a full-mission simulator or part-mission training device should be designed.

The analyst would then use the task clusters defined by the Simulation Configuration Module as the basis for the application of the Instructional Feature Selection and Fidelity Optimization Modules. These two modules would define candidate training system designs for each task cluster. The output of the two modules is a range of options that vary in cost. Thus, the overall results of the application of these modules would be a collection of training device designs specifying for each design the level of fidelity on each fidelity dimension and the collection of instructional features included in the design. The analyst would select several of these designs for further examination.

The Training Device Selection Module evaluates the training device designs produced in the previous process. The analyst would exercise this module several times using different combinations of training devices. For each combination, the module would determine the number of tasks that would be assigned to each training device, the number of hours each task would be assigned to each device to meet the training requirements at the lowest cost, and the optimal training cost given the particular combination of training devices. This model makes the simplifying assumptions that the hourly cost of a training device is fixed and that all devices are fully utilized. These assumptions allow the Training Device Selection Module to determine a solution in less than one minute.

When the analyst was relatively confident of the solution of the Training Device Selection Module, he or she would then investigate the solution using the Resource Allocation Module. It could be that the recommendations of the Training Device Selection Module would require the procurement of more training devices than would be feasible, or would provide some training on actual equipment for tasks in which such training violated safety regulations. The Resource Allocation Module allows the analyst to impose constraints on the number or use of equipment in the training system and examine the resulting optimal solution. The Resource Allocation Module also relaxes the simplifying assumptions that were used by the Training Device Selection Module to estimate training device cost, leading to a more accurate cost function. As a result of its increased generality, the Resource Allocation Module takes several minutes to reach a solution, an order of magnitude longer than the Training Device Selection Module.

At many points in the process, the analyst has the option of returning to modules that were used previously to refine the analysis, change assumptions, or choose different solutions. For example, the analyst might change the definition of the task clusters based on the results of Training Device Selection Module, or may use those results to select different candidate device designs for evaluation.

Summary of Data Requirements

All methods of training-system design require a good front-end analysis. The OSBATS model is no exception to this rule, and requires information about training requirements, task characteristics, trainee population skills, candidate training-device instructional features, and fidelity dimensions. In addition, because the model is quantitative rather than qualitative, it requires numerical estimates for many of its parameters. The OSBATS model has been designed to obtain the required data as easily as possible.

The specific data required and their formats are derived from the methods and goals of the five modules. This section presents an overview of the input data requirements, defining classes of data that are required, the required format, and potential data sources. The detailed data requirements have been enumerated by Sticha, Blacksten, Buede, Singer, Gilligan, Mumaw, and Morrison (1988) in the Model Description, Implementation, and Evaluation report.

There are two types of data required to support the functioning of the OSBATS model. The first type, called resident or internal data, covers the unchanging or slowly changing information and relational rules involved in the generation of options, tradeoffs, and configurations. The second type of data required by the model is situationally specific data, the data used to initiate execution of the models.

The resident data cover general rules for fidelity options and instructional features, fidelity and instructional feature options and cost estimates, learning parameters, and so forth. The resident data also include rules about the relationships between the resident data values and the input data. The input data are used to initiate execution of the models. These data include descriptions of the tasks to be taught, the task performance criteria to be met by the training, the number of students, and the time required for training each task.

We do not anticipate that the engineer using the OSBATS model will be the principal individual responsible for providing input data. Rather, we see two principal sources of data for the OSBATS model. First, information about the problem structure, general training-device characteristics, and inference rules will be resident in the model. This information will be updated periodically as the domain of the model expands, and as new research results or experts' opinions are incorporated into the model. The resident information should be relevant over a wide class of possible applications of the OSBATS model. The second class of data describes the specific training problem addressed by the OSBATS model. This information describes the tasks to be

trained according to the parameters of the model components. We do not expect that the user will have the subject-matter expertise required to provide these data directly. Consequently, we envision that ultimately these data will be developed through a task analysis that supports the training-device design process.

Certain inputs are required of the user, however. These inputs consist of the critical judgments that express general priorities in training-system design, and that limit the scope of the problem addressed by the OSBATS model. The user input requirements are enumerated below.

1. Weights that express the importance of operating-cost savings relative to safety and training-effectiveness concerns in determining whether a task should be trained on a training device or on actual equipment.
2. A value that reflects the importance of savings in investment cost relative to savings in operating cost.
3. Specification of whether all tasks should be weighted equally in the analysis, or whether tasks should be weighted according to estimates of the amount of training on actual equipment required to reach the training standard.
4. Specification of whether historical data on the likelihood that instructional features are used in existing training devices should be used in evaluating the benefit of these features in devices that are being designed.
5. Assumptions that should be made about training-device utilization to determine the total hourly cost of the training device.
6. Constraints on the maximum number of training devices to be procured, and on the minimum performance level in which each training device may be used for each task.
7. Limits on the tasks, training-device options, candidate instructional features, and fidelity options that are considered in the analysis.

Data Requirements and Format

We have organized the data requirements for the OSBATS modules into the following six categories with their respective subcategories.

1. Task training requirements. This class of data includes information about the training requirements associated with the tasks that must be performed to prescribed standards following training. This class includes two subclasses.

- a. Task learning points. These data describe for each task student entry performance level and performance standard on a scale that ranges from no knowledge (0) to expert performance levels (1.0).
 - b. Task simulation evaluation factors. These data include a rating of each task on a checklist of factors that are relevant to determining the need for simulation, including safety concerns, special performance conditions, and anticipated training effects.
2. Other task data. Other task data include three kinds of information about tasks.
- a. Task training hours and costs. These data describe the training time and costs involved in meeting the training requirements for each task without a training device. Data elements describe the number of training hours required in classroom, actual equipment use in both operational and non-operational modes, set-up time, and the cost of other required equipment.
 - b. Task information processing characteristics. These data rate tasks on a checklist of information-processing activities, such as timesharing or continuous-control processes, that are relevant to the evaluation of training-device instructional features.
 - c. Task activities. These data describe the activities required to perform the task according to the variables required by the fidelity rules. This class of data encompasses several variables that are specific to the task and domain.
3. Training-device data. This class of data describes hypothetical or actual training media in terms of cost, cue and response capabilities, and instructional features. This class includes three subclasses.
- a. Training-device costs. These data include the following data elements for each training device: investment cost, annual fixed operating cost, hourly variable operating cost, maximum annual utilization, and training-device life cycle.
 - b. Training-device cue and response capabilities. These data rate the technical performance of each training device on each of the fidelity dimensions defined in 4.a.
 - c. Training-device instructional features. These data provide a checklist of the instructional features possessed by each training device.

4. Fidelity dimension data. This class of data defines the set of options that are considered by the Fidelity Optimization Module, defines the technical performance scale in terms of concrete options, and contains parameters for estimating training-device cost as a function of cue and response capabilities. Fidelity dimension data include four subclasses.
 - a. Fidelity dimensions and levels. These data define each fidelity dimension and list the names of all levels and the associated technical performance rating, on a scale from 0 to 1.0.
 - b. Fidelity dimension cost data. This class of data includes the three parameters of the function that is used to estimate the cost of a particular level from its technical performance. The three parameters describe the minimum cost, maximum cost, and an exponent that describes the shape of the cost curve.
 - c. Fidelity dimension minimum performance parameter. This parameter, assessed for each fidelity dimension, estimates the transfer of training that would occur when the capability on the subject fidelity dimension is nil, but capabilities on all other dimensions are perfect.
 - d. Fidelity rules. These data are an ordered set of conditional statements that derive the task cue and response requirements from a description of the activities required to perform a task.
5. Instructional-feature data. This class of data describes the costs and benefits of the instructional features and gives specific rules for associating instructional features to tasks. Included in this class of data are two subclasses.
 - a. Instructional-feature rules. Instructional feature rules specify the conditions under which each instructional feature would improve training efficiency. The conditions may reference other elements in the data base.
 - b. Instructional-feature cost and weight. The data elements in this class include an assessment of the development cost of each instructional feature, and an assessed weight that moderates the calculated benefit values for instructional features.
6. Training-system data. This class of data includes a variety of miscellaneous data and general information about the training course. This data class includes the following two subclasses.

- a. Course and system information. A single element describing the required number of graduates per year is included in this category.
- b. Model information. This class of data includes a variety of assumptions used by the model. The nature of each data element is described in the formal model description.

Data Sources

The data required to operate the OSBATS model will come from several sources, including subject-matter experts (SMEs), training-system experts (TSEs), training researchers (TRs), model developers (MDs), and model users (MUs). As the model evolves, we expect the nature of the data required from experts to change, with subject-matter and training-system experts providing simpler judgments that are more factual and less subjective. These judgments would be transformed to produce the data required by the model. In the near term, however, experts will be required to provide a variety of judgmental data to meet the model requirements. General descriptions of these data sources are given below.

1. Subject matter experts include instructors and expert job performers. These experts are characterized by their knowledge of the tasks being trained. They are the primary source of task training requirement and other task data.
2. Training-system experts are characterized by their knowledge of the capabilities and costs of training devices. They are the primary source of training-device data, fidelity dimension data, and instructional-feature cost data.
3. Training researchers provide the link between the model and the body of relevant behavioral research. They will be the major source of instructional feature data. In addition, behavioral research will play an important part in the form of the functions that predict training cost and effectiveness.
4. Model developers are required to produce data in some of the areas in which consideration of the model form is required. For example, the fidelity optimization module assumes that cost and benefit of fidelity dimensions are mutually independent. The model developer will be able to structure the fidelity dimensions to reduce or eliminate the effect of any interactions. Consequently, we expect that the model developer will work with the training-system expert to define the fidelity dimensions and levels.

5. The model user may be one of the other four kinds of experts, or may be a project manager who must aggregate the specific expertise of staff members. Although the user will have access to the data base, we expect the user to have three major impacts: (a) to make the value judgments that affect the critical weights used at various points in the analysis, (b) to set the scope of the analysis, and (c) to adjust the results of the analysis to account for factors that are not included in the model.

System Modeling Methods

As the OSBATS model was developed, we maintained a formal system description of the model using the Integrated Computer Aided Manufacturing Definition (IDEF0) system description language, developed by the Integrated Computer Aided Manufacturing Office (ICAM) of the U.S. Air Force to be used as a tool for describing the functions and data of a complex system (SofTech, Inc., 1981; Ross & Schoman, 1977). A system consists of any combination of machinery (hardware), data, and people, working together to perform a useful function. IDEF0 is a technique that enables people to understand complex systems and to communicate their understanding to others.

We obtained several benefits from using IDEF0 to describe the OSBATS model during its development.

1. The procedures used by the model were stated explicitly and could be readily examined by sponsors, model developers, system analysts, and programmers.
2. The use of a formal modeling tool ensured that the system was complete, and that the interactions of model components were well-specified.
3. The IDEF0 model proved to be a useful tool for verification of the software. We could easily compare the code and the results of calculations to the model specifications.
4. The system model helped ensure that sponsors and members of the contractor research staff had a common understanding of the model's goals, methods, and results.

The remainder of this subsection describes the IDEF0 methodology. This description will enable the reader to understand the formal model description presented in the last part of the section.

IDEF0 describes the functions performed by the system by successively decomposing the system into its basic components, describing how each component processes information, and specifying how different components interact. An IDEF0 model is expressed as a series of related diagrams; each diagram describes a particular system component or function. An IDEF0 diagram is composed of boxes and arrows. The boxes represent component functions or activities, while the arrows represent data that affect the activities or are produced by them. In this report, IDEF0 is used to describe the components and functions of the OSBATS model.

IDEF0 Model Organization

The diagrams in an IDEF0 model describe the system in a modular, top-down fashion, showing the breakdown of the system into its component parts. The application of IDEF0 starts with the most general or abstract description of the system to be produced. This description is represented in a diagram as a single box; that box is subsequently broken down into a number of more detailed boxes, each of which represents a component part. The component parts are then detailed, each on another diagram. Each part shown on a detail diagram is again broken down, and so forth, until the system is described to the desired level of detail. Lower-level diagrams, then, are detailed breakdowns of higher-level diagrams. At each stage of breaking down the system, the higher-level diagram is said to be the "parent" or overview of the lower-level "detail" diagrams. The relationship between diagrams at different levels is shown in Figure 2.

Diagram display format. In this document, each diagram in an IDEF0 model is displayed in a two-page format. The subject diagram is shown on the top of the right-hand page. The parent of the subject diagram is shown on the top of the left-hand page with the location of the subject node indicated. On the bottom half of both pages is text describing the operations performed by each activity represented in the diagram. Each pair of pages receives a page number that is displayed as part of the subject diagram.

Diagram node numbers. In an IDEF0 diagram, the component parts are shown as numbered boxes. A diagram should have no more than six boxes. Each box at one level is detailed in one diagram at the next lower level until a sufficient level of detail is reached. The place of each diagram in a model is indicated by a "node number" derived from the numbering of boxes. For example, A21 is the diagram that details box 1 on the A2 diagram. Similarly, A2 details box 2 on the A0 diagram, which is the top diagram of the model. The parent of the A0 diagram represents the system as a single box and is denoted "A-0." The hierarchy may be shown in an index of diagram names and their node numbers called a "node list." The node list serves as a table of contents for a model. In an IDEF0 model, diagrams are displayed according to the order of their node numbers.

The example shown in Figure 3 provides an illustration of the hierarchical decomposition of functions. The diagrams in Figure 3 indicate that the overall function, develop system (A0), is broken down into three sub-functions, A1 through A3. Design system (A2) is further broken down into three, more detailed sub-functions (A21 through A23).

Description of Individual IDEF0 Diagrams

In IDEF0, boxes represent activities required to perform a function, and arrows represent relationships between these activities. Descriptive labels are written inside each box and along each arrow to describe their meaning. The notation is kept simple to permit easy reading with little special training.

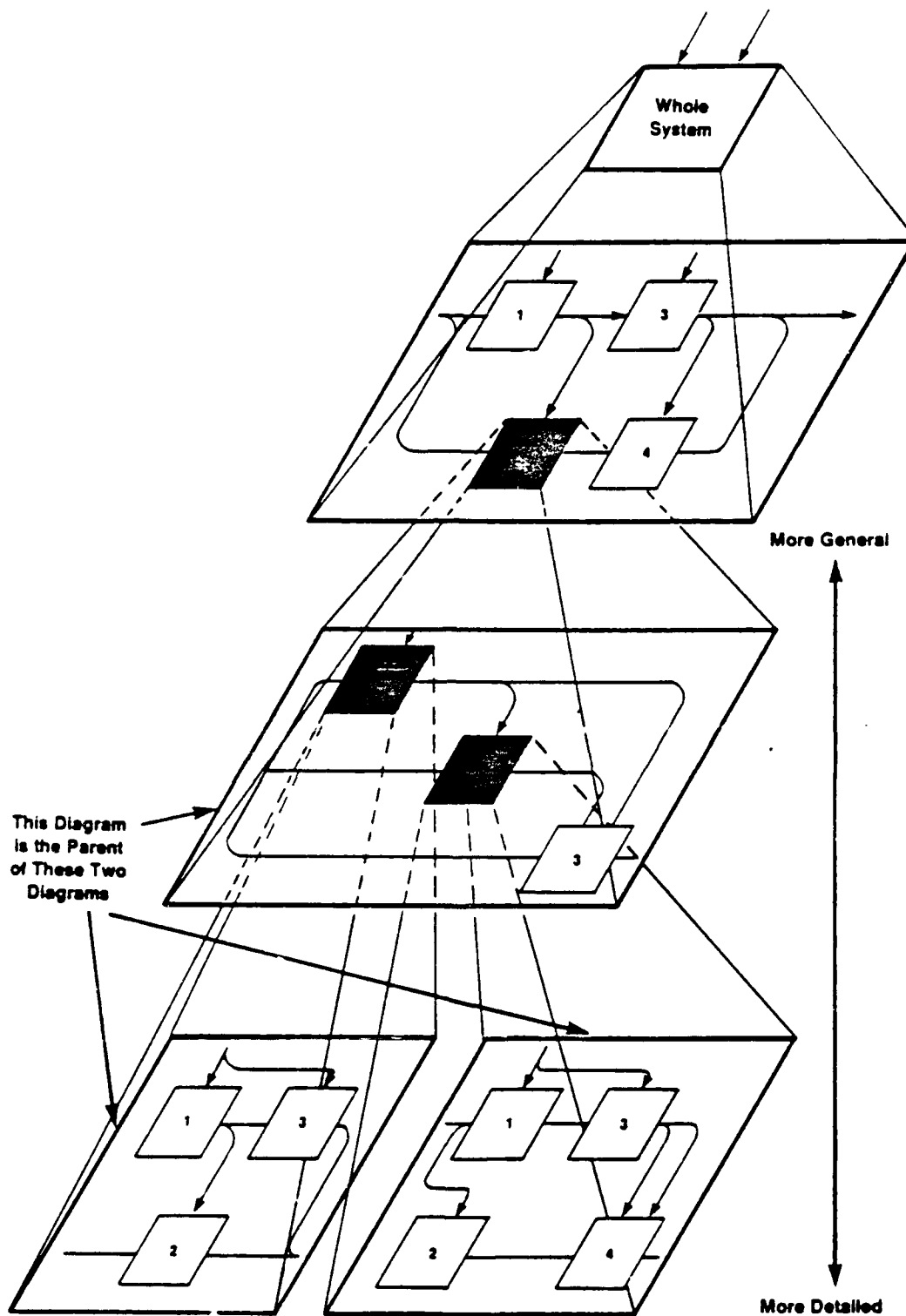


Figure 2. Example of a hierarchical, top-down model.

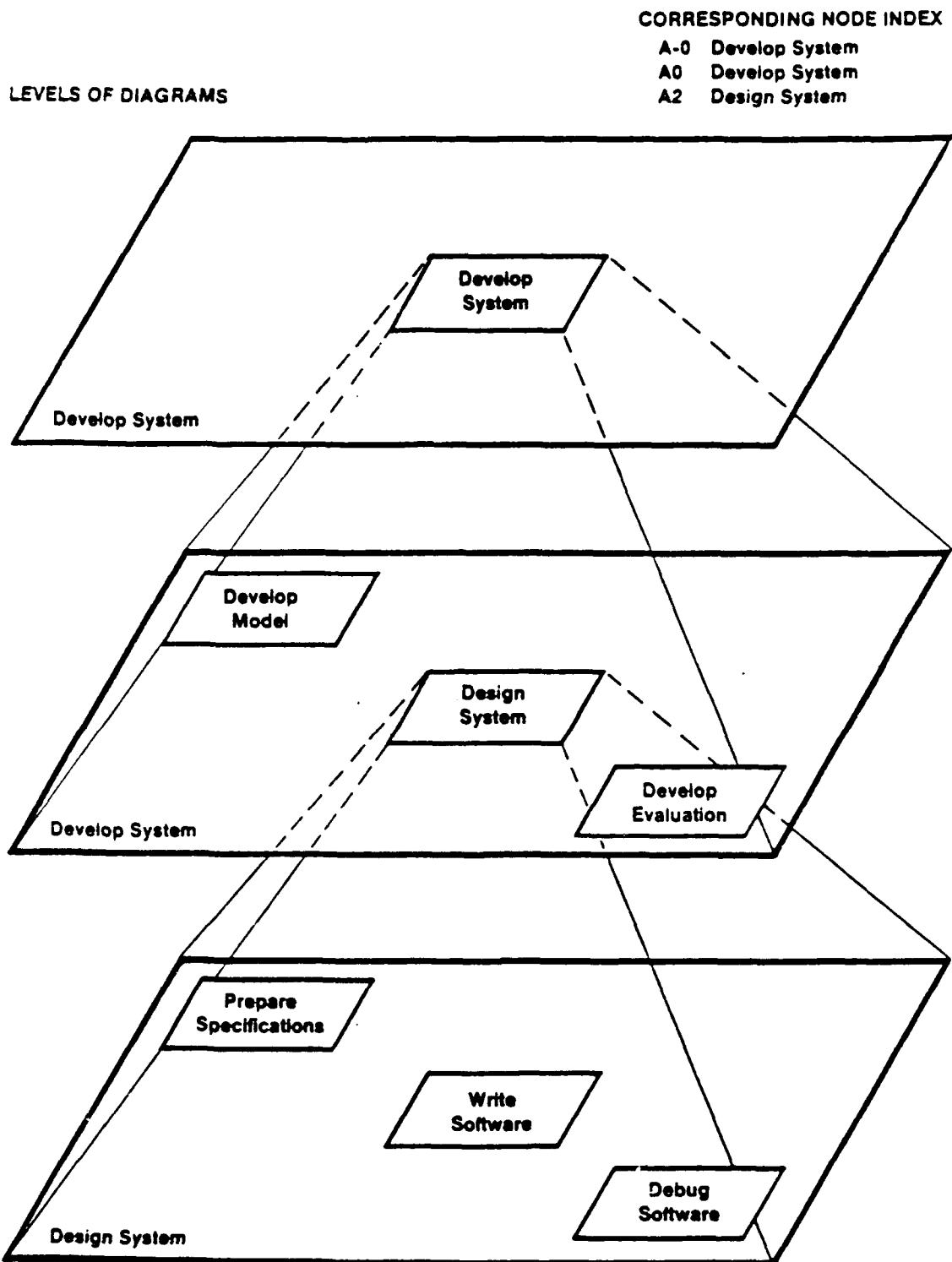


Figure 3. IDEF0 node numbering convention.

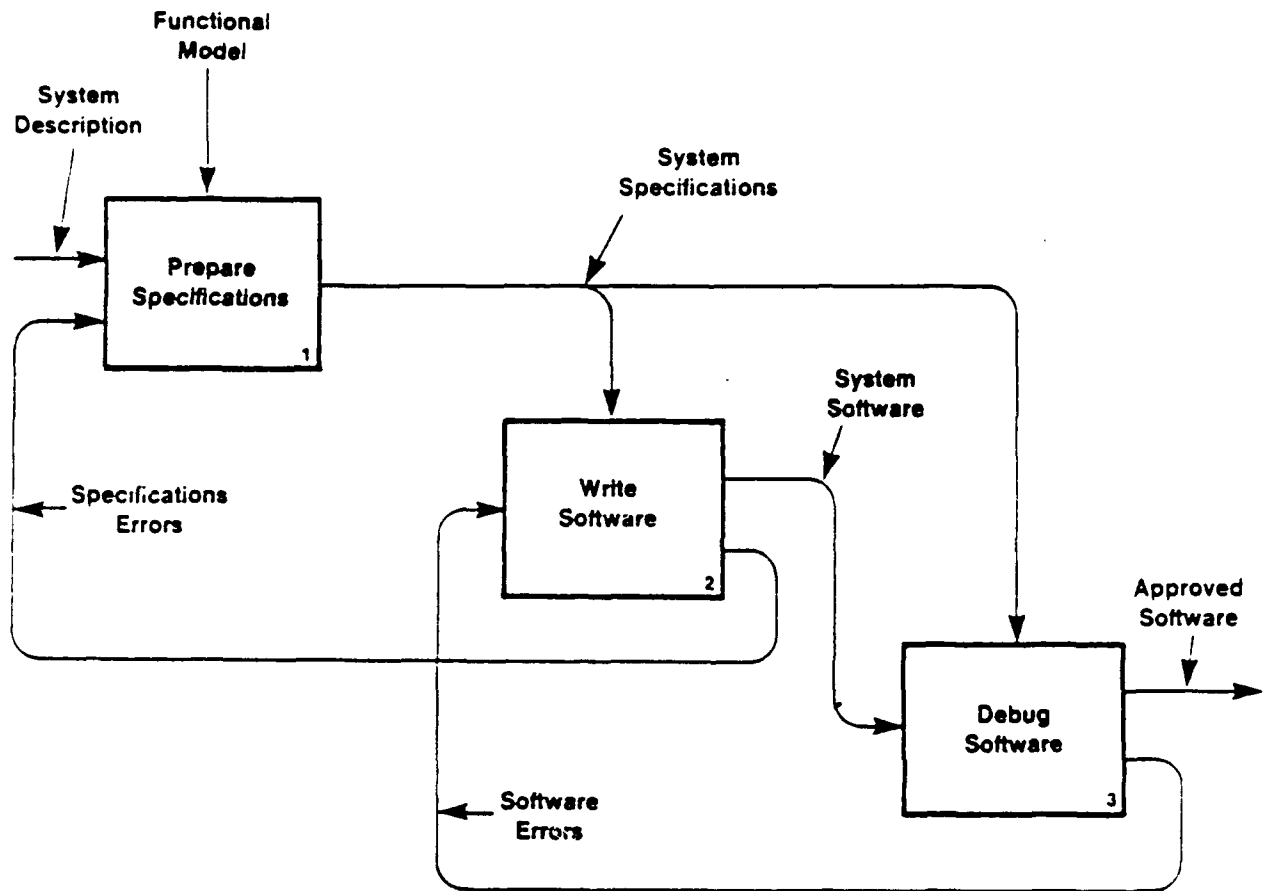


Figure 4. Sample IDEF0 diagram.

Figure 4 shows a sample IDEF0 diagram. Notice that the boxes represent the breakdown of activities or functions performed by the system and are named by verbs. Arrows, which represent objects or information, are labeled with nouns.

Box-and-arrow syntax. The sample IDEF0 diagram in Figure 4 shows that the descriptive names and labels convey the box and arrow contents to the reader. In addition to its label, the side at which an arrow enters or leaves a box shows its role as an input, control, output, or mechanism for the box (see Figure 5). Arrows that enter from the left of an activity box are inputs to the process represented by the box. Inputs represent the raw materials or data used by the activity to produce outputs. The outputs are represented by arrows that originate from the right side of the box. Arrows entering a box from the top are controls on the activity. Controls are data that provide catalysts or constraints for the represented activity, but are not changed by the process. Finally, arrows that enter a box from the bottom

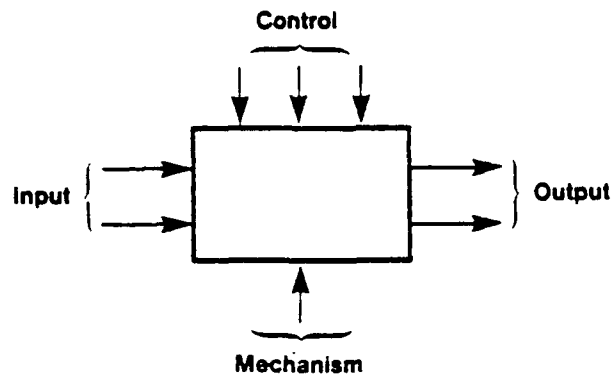


Figure 5. Sample IDEF0 diagram showing box and arrow syntax.

represent mechanisms. Mechanisms are the agents that perform the activities represented in the box. In short, inputs and outputs represent what is done by the process, controls represent why it is done, and mechanisms represent how it is done.

The arrow structure of an IDEF0 diagram represents a constraint relationship among boxes. It does not represent flow of control or sequence. The arrows entering a box show all that is needed by the box to perform its function. Therefore, the box is constrained by its inputs and controls.

Labeling of arrows. Some arrows show both their source and destination boxes on the same diagram, while others have one end unconnected (see Figure 6). The unconnected arrows represent inputs, controls, or outputs of the parent box. To find the source or destination of these unconnected arrows, the reader must locate the matching arrows on the parent diagram. All such unconnected arrows must continue on the parent for the diagrams to be complete.

Although arrow connections from parent boxes to detail diagrams are sometimes obvious from the labels, we have developed a special notation that should allow readers to do the match quickly. The notation used to describe the OSBATS model is slightly different from standard IDEF0 procedures for labeling unconnected arrows. The data for the OSBATS model is described in a structured data base. Each element in the data base is identified by a unique outline number (e.g., 2A1). Input and control arrows that represent data in the data base are labeled with the appropriate outline number. Often data are described more generally at higher-level nodes than they are at lower-level nodes. Thus, a particular input or control may be labeled "2" at node A0, "2A" at node A1, and "2A1" at node A13.

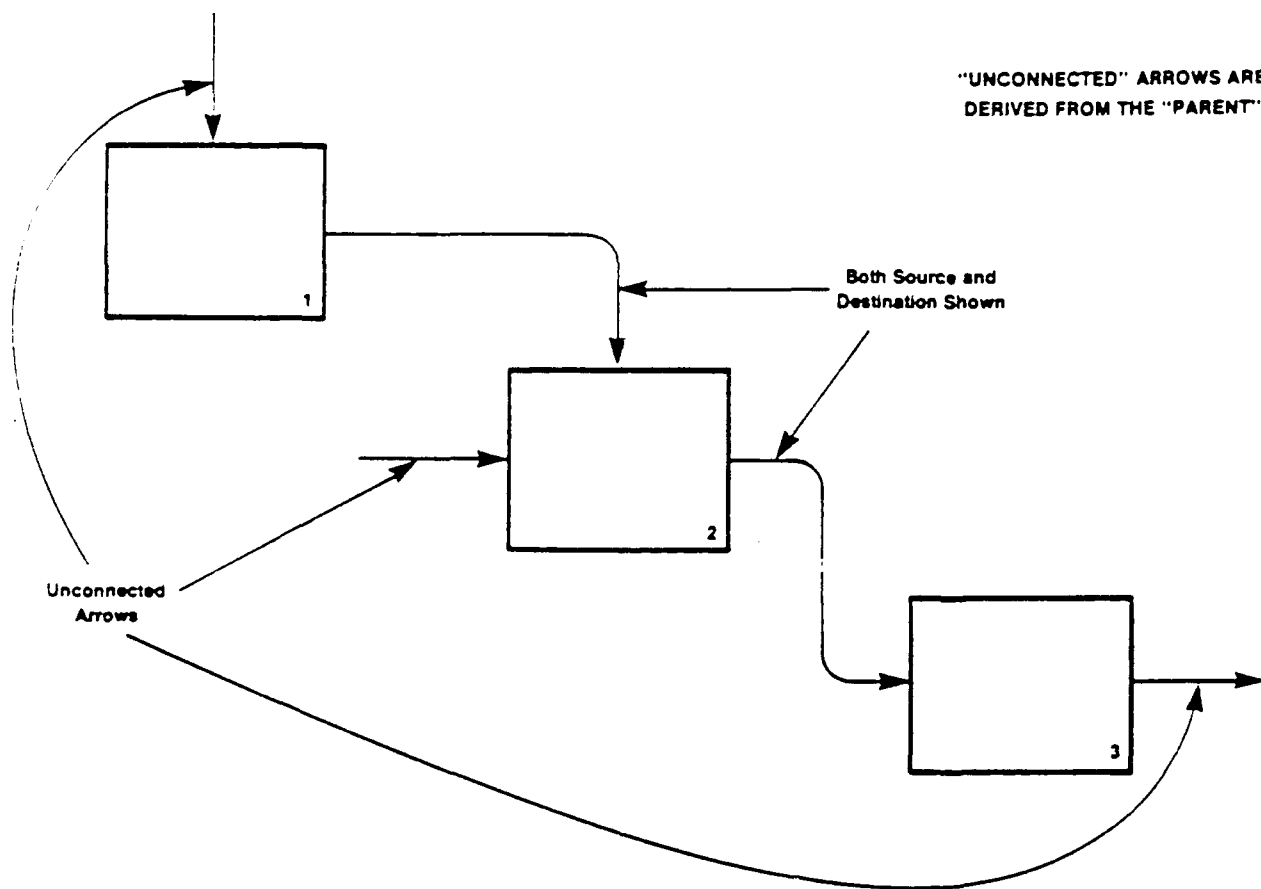


Figure 6. Sample IDEF0 diagram showing source and destination.

A somewhat different labeling scheme is used for output arrows. Output arrows are labeled according to the highest-level node at which the output originates. For example, the output of node A212 will be labeled O212A if it does not occur at any higher-level node. If there are three outputs for A212, they will be labeled O212A, O212B, and O212C. The label is consistent across all nodes in which the output is represented. Therefore, if the first output for node A212 is also shown at node A21234, it will still be labeled O212A. Occasionally, the same output is represented at higher- and lower-level nodes, but it is more detailed at the lower-level node. When this occurs, the output will retain the node number of the higher-level node but will receive an additional number to represent the division of the output into parts. For example, the output O212A may be represented as O212A1, O212A2, and O212A3 at a lower-level node. If one of these outputs is further subdivided at a lower-level node, it will receive a second letter. For example, if O212A2 is divided into three components, the components will be labeled O212A2A, O212A2B, and O212A2C.

Mechanism arrows are used sparingly in the OSBATS model definition. When they are used, their reference is clear. Consequently, the mechanism arrows are not numbered and are identified only by their label.

It is possible for a data element to serve as an input to some sub-activities of a given activity and as a control for other sub-activities. In this case, the data will be represented once in the parent diagram, either as input or control. In the detailed diagrams, the data would be represented as a control in some diagrams and as an input in others, as appropriate.

OSBATS System Model

This section presents an overview of the IDEF0 description of the OSBATS model. The complete system description is presented in the Model Description, Implementation, and Evaluation Report (Sticha, Blacksten, Buede, Singer, Gilligan, Mumaw, and Morrison, 1988). The description begins with a list of the nodes in the system model in the order that they appear in the system description. The node list provides the table of contents for the IDEF0 model. If the node is represented by its own diagram, the number of that diagram is listed in the final column of the node list. Nodes that have no detailed diagram do not have an IDEF number listed. The descriptions of such a node may be found on the diagram for its parent node.

A description of a single node in the model consists of three components: two diagrams (which may be repeated) and associated explanatory text. The diagram for the node being described is on the right-hand side of the page; its parent is shown on the left-hand side. The text is written beneath the diagrams. If the explanatory description requires more than two pages, both parent and child diagrams are repeated on the next two pages, until the text is completed.

Table 1. IDEF0 Node List

Node	Title	IDEF Nr.
A-0	Optimization of Simulation-Based Training Systems	1
A0	Optimize Training System Design	2
A1	Perform Preliminary Processing	3
A11	Analyze Tasks for Instructional Features	
A12	Analyze Tasks for Fidelity Features	
A13	Determine Task Weights	
A14	Determine Learning Function	
A2	Develop Training Concept	4
A21	Recommend Simulator Configuration	5
A211	Evaluate Each Task	
A212	Develop Simulation Recommendations	
A3	Design Training Devices	6
A31	Select Instructional Features	7
A311	Select Tasks and Candidate Features	
A312	Calculate Benefits of Features	
A313	Select Optimal Features	
A32	Optimize Device Fidelity	8
A321	Construct Training Device Options	
A322	Calculate Costs and Benefits of Options	
A323	Compute Optimal Device Designs	
A4	Assign Training to Devices	9
A41	Select Training Device for Tasks	10
A411	Determine Training Device Hourly Cost	
A412	Identify Cost Effective Devices	
A413	Examine Device Utilization	
A42	Allocate Training Resources To Training Devices	11
A421	Detail Cost Curves	
A422	Solve Multi-Task Resource Allocation Problem	
A423	Check Solution Location Vis-a-Vis Detailing	

OPT/A-0: Optimization of Simulation-Based Training Systems (OSBATS)

The goal of the OSBATS Model is to specify the designs and concepts of use for training devices to meet training requirements at the least cost, or provide the greatest training effectiveness, given the cost. Training effectiveness is measured by the level of performance on actual equipment for relevant tasks resulting from the use of the training system. Cost comprises investment, fixed operating, and variable operating components.

This analysis breaks up the overall problem of training system design into three subproblems, and describes tools that can be used to address these subproblems. The diagrams describe OSBATS activities from the viewpoint of the model developer. That is, they show the data required by model processes, and describe in detail the relationships presumed by the model.

The IDEF0 analysis assumes that training requirements are defined by a set of tasks that must be performed to prescribed standards following training. Tasks are rated along several dimensions prior to the application of the OSBATS model. Training devices are characterized by their capability to present cues and collect responses, and by the instructional support features they possess. The population of instructional features and cue/response dimensions is assumed to be known prior to the application of the model, and is represented in the model data base. Training devices evaluated by the OSBATS model may represent currently existing devices, or they may be generated by the design components of the OSBATS model. For the purposes of this analysis, these two types of training devices are not separated; that is, there is a single list of training device candidates that combines existing devices with devices designed by the model.

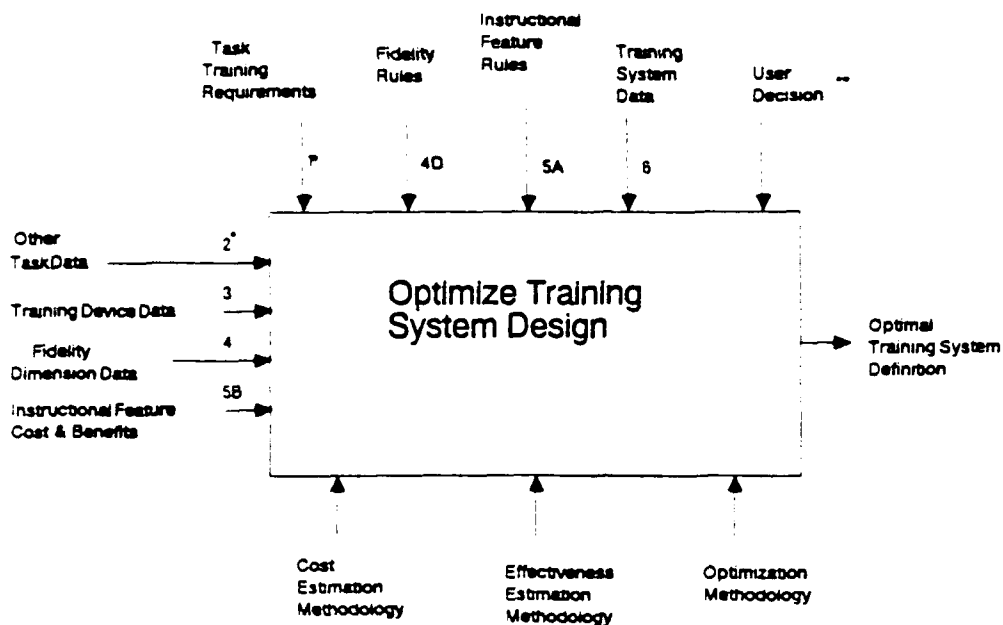
The model considers six kinds of input and control data: (1) Task training requirements, (2) Other task data (3) Training device data, (4) Fidelity dimension data, (5) Instructional feature data, and (6) Training system data. Data from each source are used in one or more model component.

Task training requirements describe the tasks to be trained, specifying entry performance level and training standards, safety and special performance conditions.

The three components of other task data describe (1) the training time that would be required to meet the training requirements in one hypothetical case, the case in which all training is conducted in a classroom or on actual equipment, (2) task information-processing characteristics, and (3) task activities related to cue and response requirements.

Training device data characterize the hypothetical and actual media that might be used to provide training. These data describe the costs, cue and response capabilities, and instructional features present in each alternative device.

Fidelity dimension data characterize the ways that simulators or other training devices may differ in the accuracy with which they present the stimuli and response options from the actual equipment and operating environment in which the tasks are



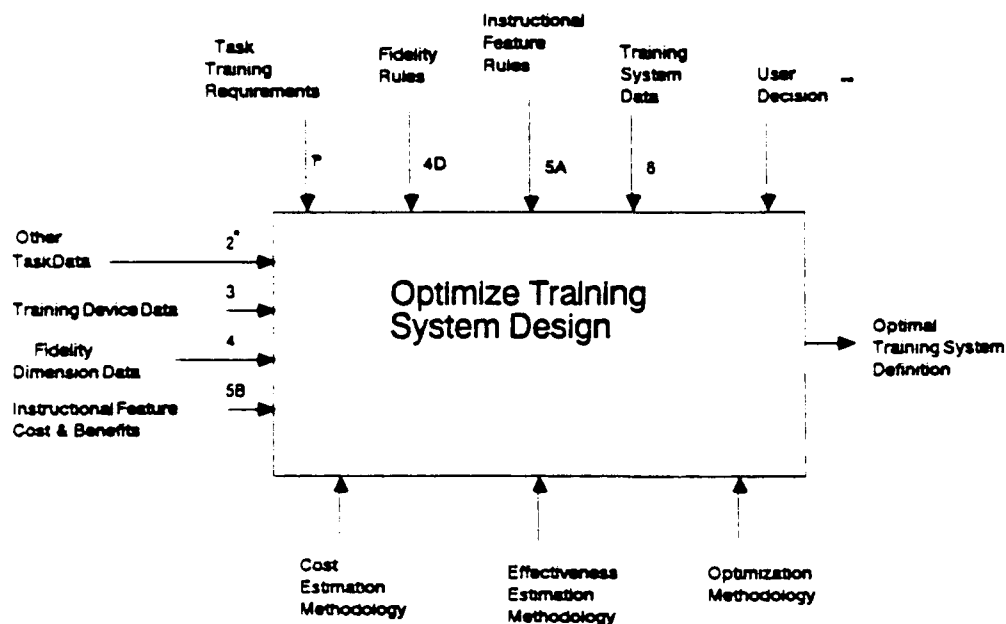
Node: OPT/A-0	Title: Optimization of Simulation-Based Training Systems (OSBATS)	Date: 8/1/88	Number: OPT-1
------------------	---	--------------	---------------

- * Labeled inputs and controls refer to elements contained in a database.
 ** Unlabeled inputs and controls refer to user-supplied information.

performed. These data are used to design training devices that provide the optimal levels of fidelity on each fidelity dimension as a function of training device cost.

Instructional feature data include the information required to select the instructional features that provide the maximum improvement in training efficiency as a function of cost. This segment of the data base includes rules used to select instructional features, and data used to calculate cost and benefit of instructional features.

Finally, training system data include general information about the training system, and other miscellaneous information required by the model. Specific data include the annual requirement for graduates and several assumptions used by the model.



Node: OPT/A-0	Title: Optimization of Simulation-Based Training Systems (OSBATS)	Date: 6/1/88	Number: OPT-1
------------------	---	--------------	---------------

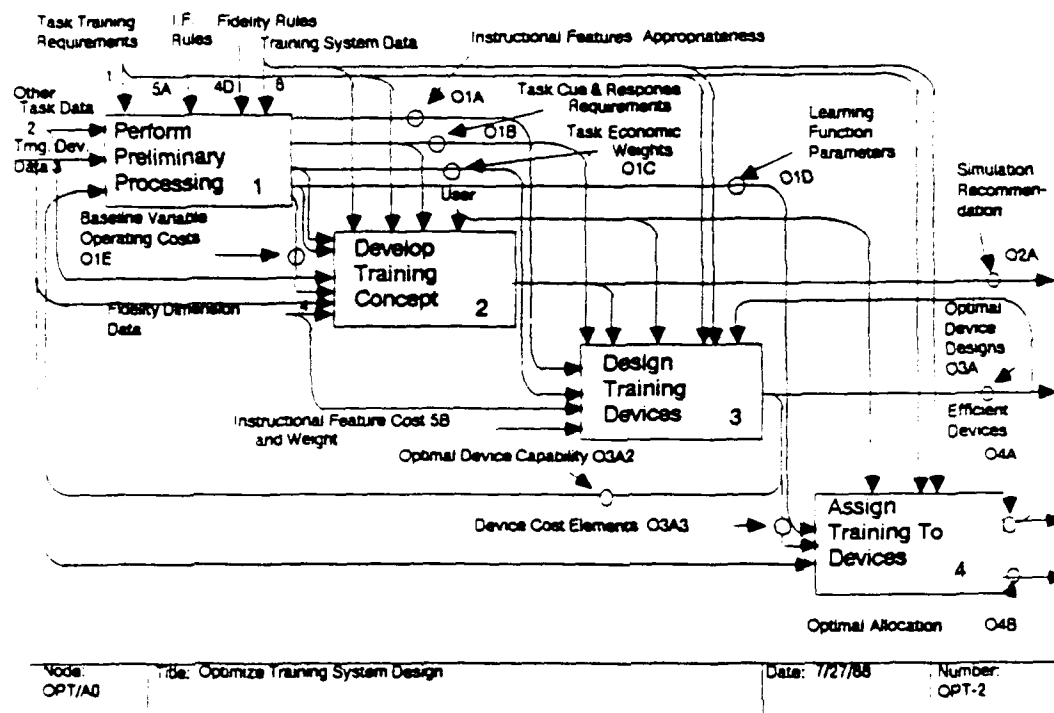
- * Labeled inputs and controls refer to elements contained in a database.
 -- Unlabeled inputs and controls refer to user-supplied information.

OPT/A0: Optimize Training System Design

The first level of decomposition describes the three problem areas addressed by the model: (1) training-concept development, (2) training device design, and (3) assignment of training resources to devices. Since common processes are involved in several of these problems, the analysis includes a fourth component that conducts the preliminary processing required for more than one model component.

OPT/A1: Perform preliminary processing. This activity produces the following basic information that is processed further by several model components: (1) A task-by-instructional-feature matrix that specifies which instructional features would enhance training for each task, (2) a matrix that describes the cue and response requirements of each task along each fidelity dimension, (3) task weights that reflect the relative cost of training each task on actual equipment, and (4) parameters that describe the course of learning for each task on each candidate training device. Although this activity is preliminary to the operation of any specific tool, the node does receive feedback from the training-device design process in node OPT/A3. That is, devices designed in OPT/A3 will need to be processed by this node to provide the data required by other modules.

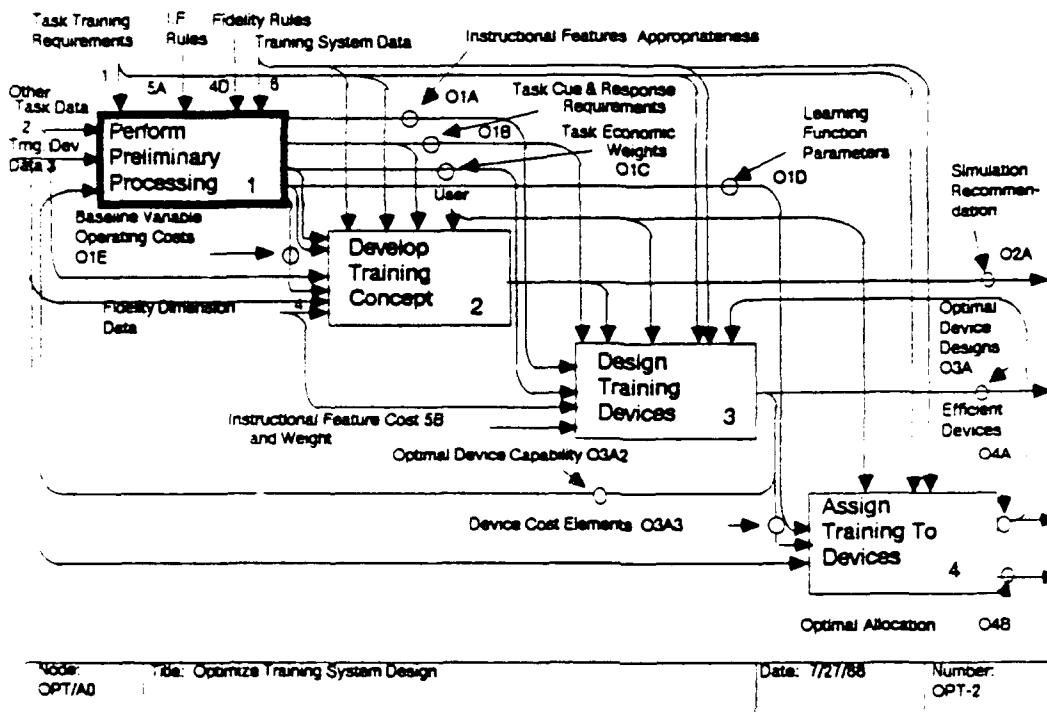
OPT/A2: Develop training concept. This problem area is concerned with finding clusters of tasks that have similar training-device needs. These tasks form



training requirements to be used in the training-device design process (OPT/A3). Training-concept development occurs early in the training system development process, and is refined several times during this process. The OSBATS model contains a single tool for training-concept development. This tool evaluates different classes of training devices (full-mission simulators, part-mission simulators, actual equipment) that may meet parts of the training requirements.

OPT/A3: Design training devices. This problem area is concerned with designing training devices that provide the optimal training effectiveness given the investment cost. The principal problems addressed in the model are (1) providing the optimal level of fidelity with respect to presentation of the environment and equipment, and (2) selecting instructional features that are tailored to the tasks being trained on the training device.

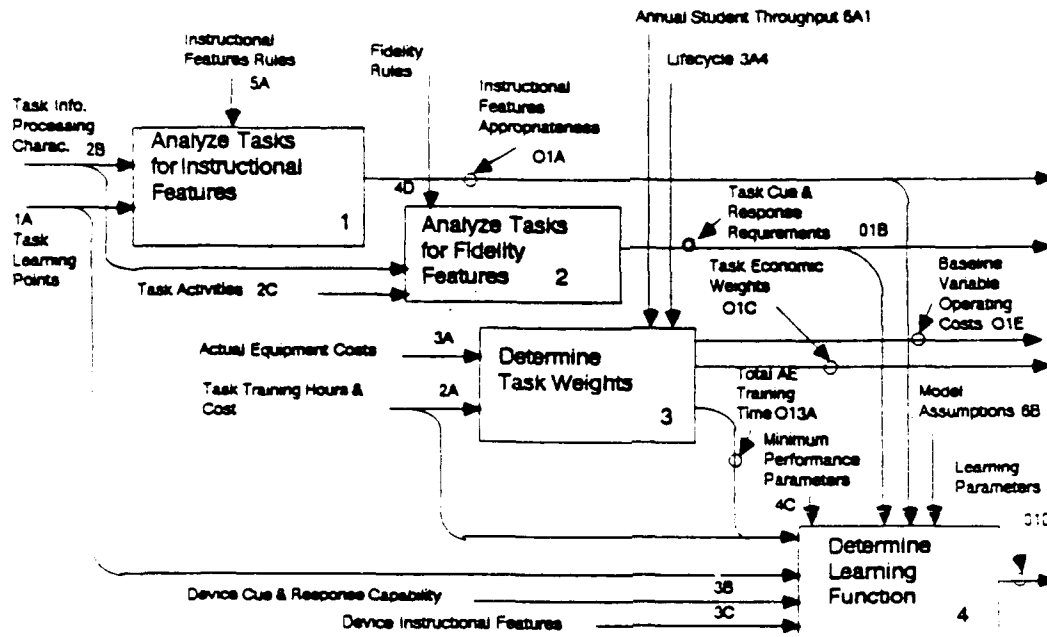
OPT/A4: Assign training to devices. This problem is concerned with determining which training device or devices should be used to train each task, and how much training should be conducted on each device. Two tools have been defined for this problem. The training-device selection tool bases the allocation on some simplifying assumptions that provide for simpler, interactive operation. The resource allocation tool provides a more detailed allocation of training resources to training devices that relaxes some of the simplifying assumptions used by the Training Device Selection Module.



OPT/A1: Perform Preliminary Processing

This activity calculates four sets of variables that are used by other model components. The first subactivity applies a set of rules (5A) that examine characteristics of the tasks to identify effective instructional features. The second subactivity applies a set of rules (4D) to identify task cue and response requirements. The third subactivity determines the total cost to train each task on actual equipment, and normalizes this cost to be used as a task weight. The fourth activity uses task and training-device data to estimate the parameters of training-device learning functions for each task.

OPT/A11: Analyze tasks for instructional features. This activity determines the instructional features that are appropriate for each task. The instructional feature rules (5A) specify the kinds of tasks for which each instructional feature is appropriate. The conditions of these rules are compared to the characteristics of the tasks (2B and 1A) to specify for each task the set of applicable instructional features. The output of this activity is a matrix, $[IFT_{T_k}]$, that indicates whether instructional feature k will enhance the training efficiency of task T . An element IFT_{T_k} of this matrix contains the value 1.0 if instructional feature k is appropriate for task T . This activity is conducted as follows.



Node: OPT/A1	Title: Perform Preliminary Processing	Date: 7/27/88	Number: OPT-3
-----------------	---------------------------------------	---------------	------------------

First, a rule base is interrogated to compare task data and rule conditions. The instructional feature rules (5A) associate instructional features with specific task characteristics. An example of such a rule is given below:

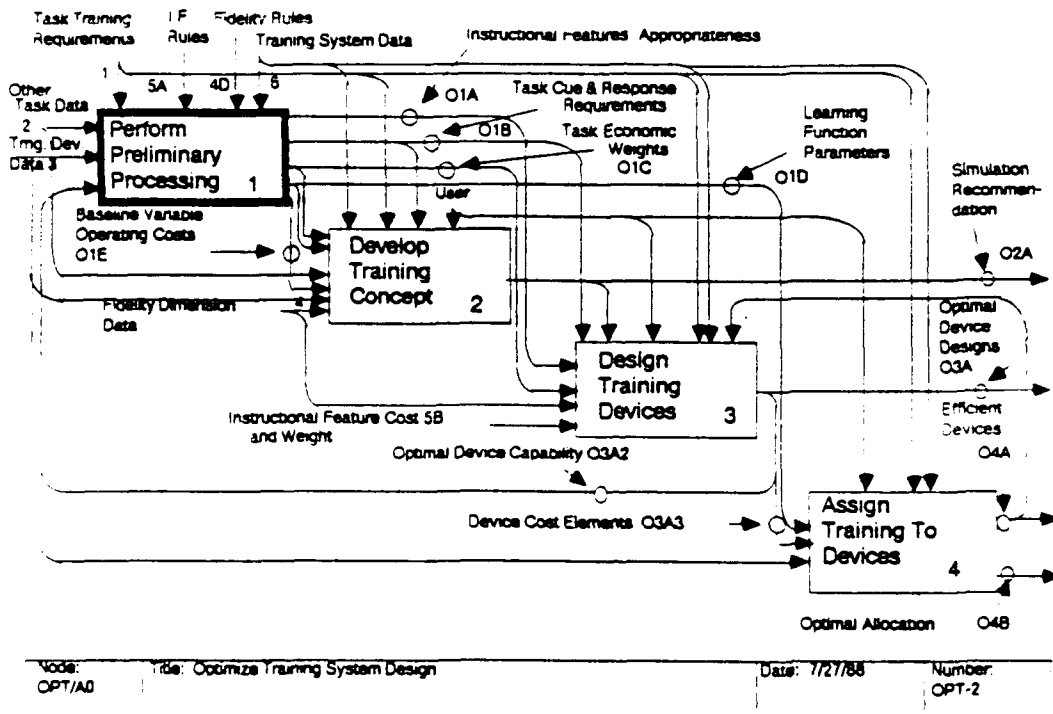
IF: Entry performance (1A1) < 0.4,
 and Intrinsic feedback (2B11) is absent,
 and The task involves continuous movement (2B1),
 or procedures (2B2),
 or decision making/rule using (2B4),

THEN: Automated Performance Alerts is indicated for this task.

This activity compares the conditions of the instructional feature rules to the task characteristics (2B) and task learning points (1A), and identifies matches, which it passes on to the next activity.

Next, this activity takes the matches produced in the interrogation, and sets the corresponding cells of IFT_{TK} . The IFT matrix is defined as follows:

$$IFT_{TK} = \begin{cases} 1 & \text{if a match was found between feature } k \text{ and task } T \\ 0 & \text{otherwise.} \end{cases}$$

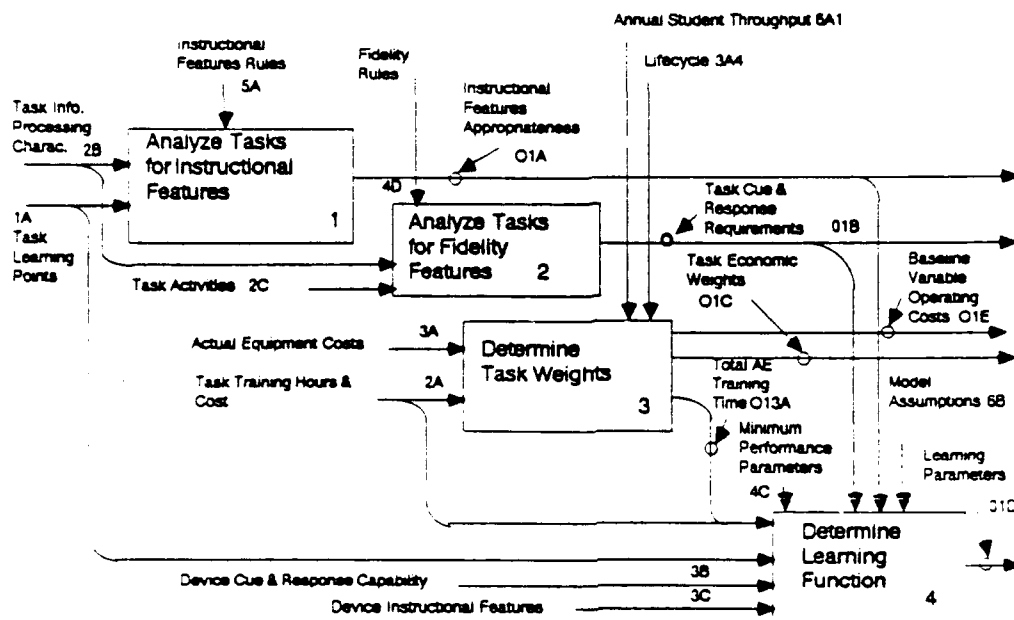


OPT/A12: Analyze tasks for fidelity features. This activity derives the task cue and response requirements (FRQT_n) on each fidelity dimension. The derivation is controlled by the information in the fidelity rule base (4D). The fidelity rule base calculates the FRQT_n based upon the values of variables that describe task activities (2C) and information processing requirements (2B). The task activity variables are situation and domain dependent. The rule base is hierarchically organized and operates using backward chaining.

An example of one of the rules in the rule base is the following:

IF: Performance cues are provided by longitudinal acceleration,
 and Motion cues should be provided by platform motion (as opposed to seat motion),
 and The magnitude of the longitudinal acceleration cues is moderate or high,
 and The task requires the performance of an emergency procedure,
 and The platform longitudinal acceleration provides a cue for the initiation of the emergency procedure

THEN: Platform surge is required for training (the requirement for platform motion is given the value 0.9).



Node: OPT/A1	Title: Perform Preliminary Processing	Date: 7/27/88	Number: OPT-3
-----------------	---------------------------------------	---------------	------------------

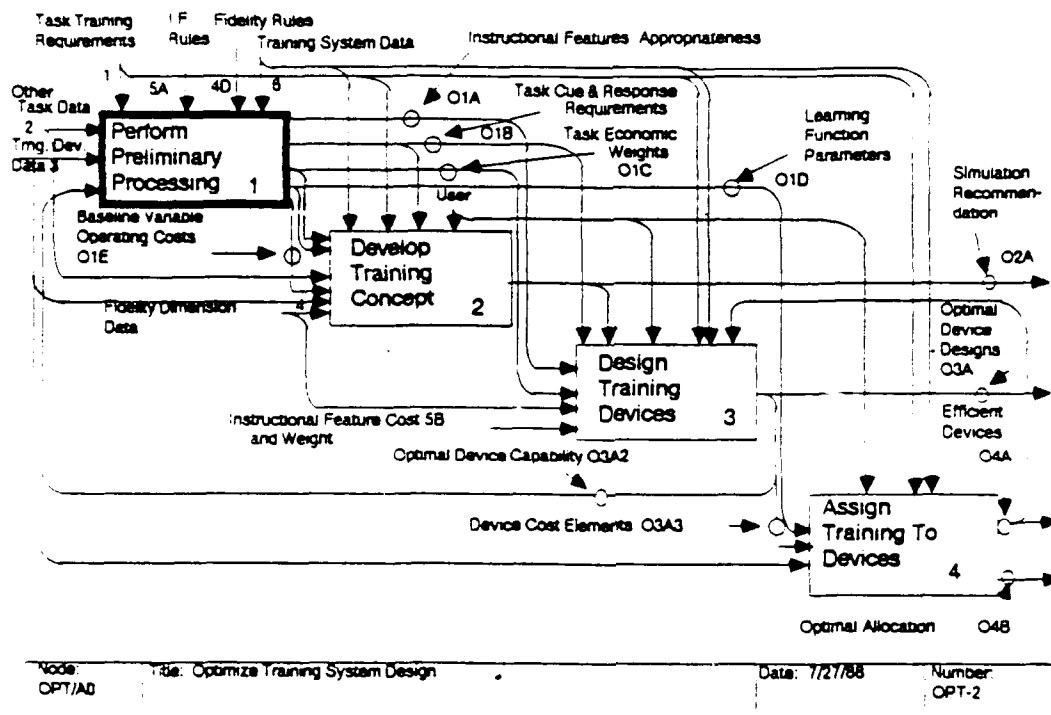
Some of the conditions for this rule are derived from the value of other data; other conditions are direct data values. The resulting values of $FRQT_{\pi}$ are in the interval $[0, 1]$.

OPT/A13: Determine task weights. This activity calculates a set of weights that are used to compare the importance of training improvements for different tasks. The weights are based on the cost of training the tasks on actual equipment. The processes contained in this activity calculate the investment (OPT/A131), fixed operating (OPT/A132), and variable operating (OPT/A133) components of the cost of training on actual equipment. The final process (OPT/A134) combines and normalizes the cost estimates to produce task weights.

OPT/A14: Determine learning function. The basis for the determination of training effectiveness is a learning function that relates performance, $P_{\pi}(t)$, to task and device variables. We use a power function of the following form to describe learning:

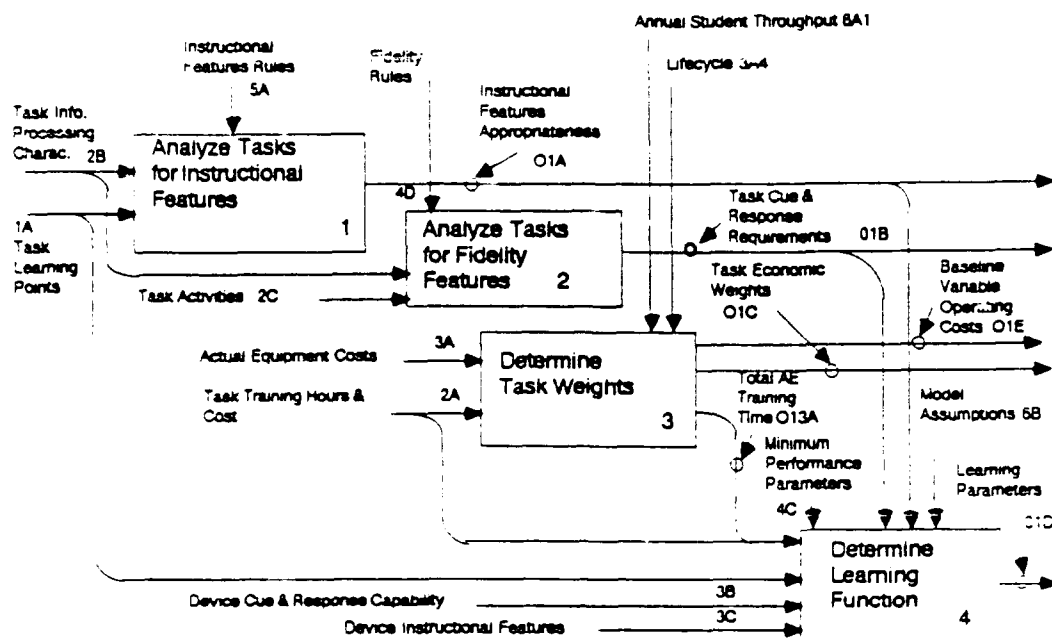
$$P_{\pi}(t) = ASM_{\pi} \{1 - [1 + TM_{\pi}TSF_{\pi}(HS_{\pi} + t)]^{\gamma}\},$$

where

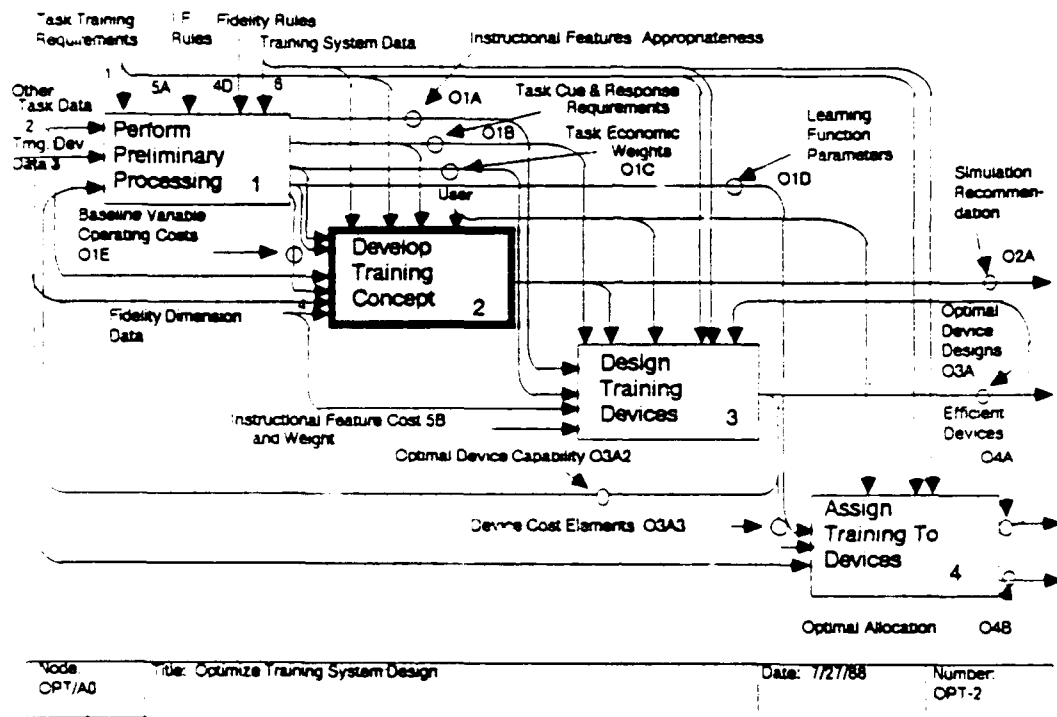


t = the training time,
 T = the task to be trained,
 i = the training device,
 ASM_T = the asymptote of the learning curve,
 TM_T = the time multiplier for the particular device and task,
 TSF_T = the time scaling factor for the task,
 HS_T = the head start for the device and task, and
 r = the learning curve exponent.

The power function form is used because of the good fit such a function has provided to empirical data (Newell & Rosenbloom, 1981). For actual equipment the parameters of this function may be based on fits to empirical research or on fits to learning curves estimated by subject matter experts (SMEs). For notional equipment, we have developed a procedure (OPT/A14) for estimating the learning curve parameters by extrapolation. The processes in this activity first calculate some of the basic parameters needed to calculate the asymptote of the learning function (OPT/A141). Then specific processes estimate the values for the parameters that do not depend on the training device (OPT/A142), and those that do depend on the training device (OPT/A143), respectively.



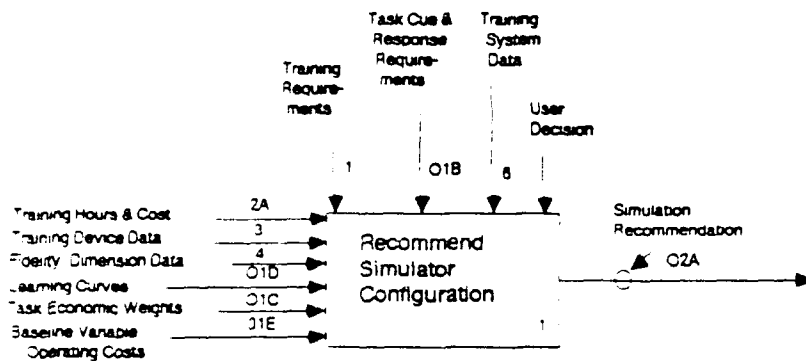
Node: OPT/A1	Title: Perform Preliminary Processing	Date: 7/27/88	Number: OPT-3
-----------------	---------------------------------------	---------------	------------------



OPT/A2: Develop Training Concept

This component is concerned with methods to cluster tasks that have similar training-device needs. The output of this component is a set of preliminary training-device requirements that can be used as the basis of the training-device design process in OPT/A3. We have currently developed a single tool for this activity, a tool that evaluates general training system alternatives including full-mission simulators, part-mission simulators, and actual equipment. Because there is only one tool in this component, this level of decomposition in the IDEF0 model is not required. However, it was included for two reasons. First, adding this level allows us to represent all tools at the same level of decomposition in the overall model. We hope that this parallel structure will make the numbering of model processes easier to understand. Second, including this level of decomposition provides a place holder for future tools that might be developed for training-concept development.

OPT/A21: Recommend simulator configuration. This tool evaluates general training system alternatives that tasks can be assigned to. The alternatives considered describe the classes of training device that could be used to provide training. Three basic classes of device are considered: (1) full-mission simulators, (2) part-mission simulators, and (3) actual equipment. The tool evaluates these alternatives, and provides additional guidance regarding the types of devices that would be appropriate given the training requirements. The recommendations are



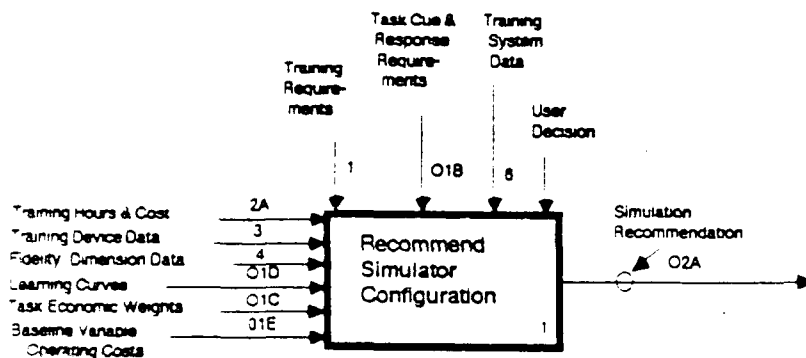
Node:
OPT/A2

Type: Develop Training Concept

Date: 7/12/88

Number OPT- 4

based on the task requirements and estimates of the cost to meet these requirements. The activity contains two subactivities, evaluate each task (OPT/A211) and develop simulation recommendations (OPT/A212).

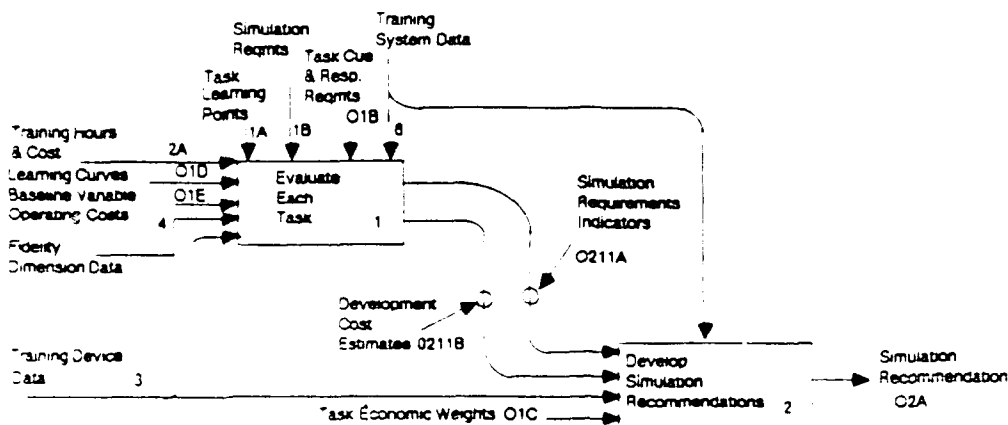


Node: OPT/A2	Title: Develop Training Concept	Date: 7/12/58	Number OPT- 4
-----------------	---------------------------------	---------------	---------------

OPT/A21: Recommend Simulator Configuration

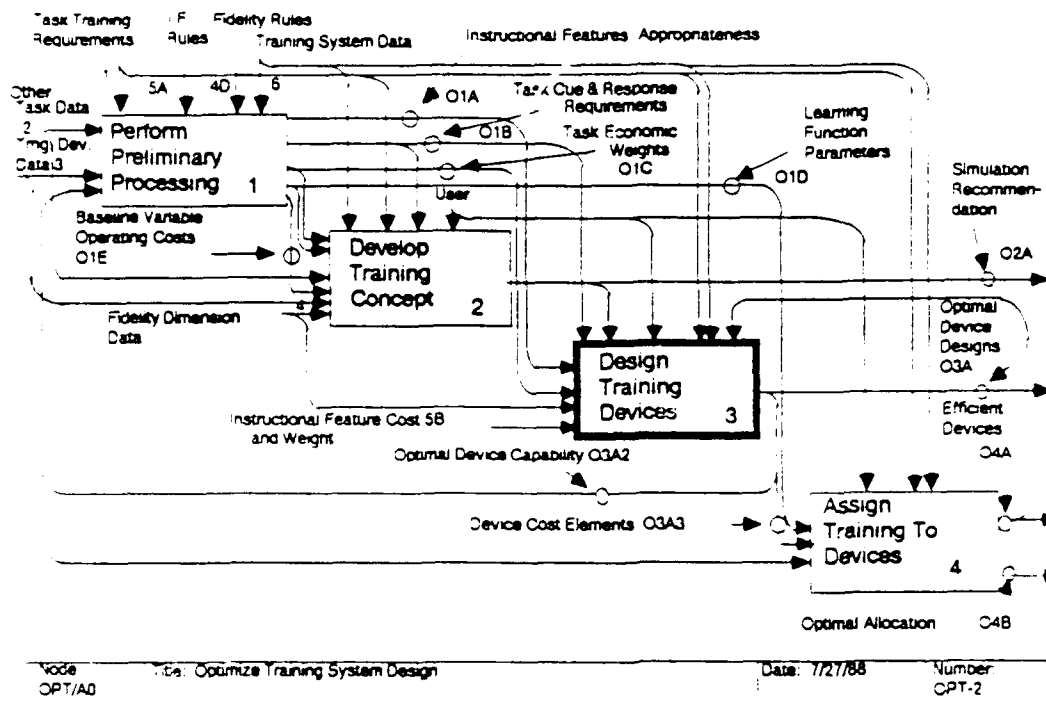
This tool examines the need and cost-effectiveness of using either a full-mission simulator (FMS) or one or more part-mission simulators (PMSs) to replace training time in the actual equipment. Built into the evaluation performed by this tool is the examination of simulator-unique capabilities, such as training in unsafe situations, and cost savings to establish the value of training with some sort of training device. In addition, the development cost of the training device that would be required to achieve these benefits is examined. Using these results the module partitions the set of training requirements into the following three subsets: (1) a subset for which an FMS should be designed by the later modules, (2) a subset requiring one or more PMSs to meet the training needs, (3) a subset requiring training on actual equipment.

OPT/A211: Evaluate each task. This first process is a set of operations that acts upon each task; the operations under this first process describe what is done for an individual task; they are repeated for every task. The requirements for training by simulator and the operating cost savings associated with simulator training (O211A) are first examined for each task (in OPT/A2111). Then the task cue and response requirements to achieve these results are used to provide a preliminary estimate of training-device development costs (O211B; evaluated in OPT/A2112).



Node: OPT/A21	Title: Recommend Simulator Configuration	Date: 7/12/88	Number: OPT- 5
------------------	--	---------------	-------------------

OPT/A212: Develop simulation recommendations. This second process then operates on the simulation requirement index (O211A) and development cost index (O211B) from the previous process to generate a recommendation concerning the need for a full-mission simulator versus several possible part-mission simulators. This recommendation compares the development cost index with the potential operating cost savings over the equipment lifecycle to recommend simulation-based or actual-equipment-based training. The development cost index is used to recommend training on an FMS or PMS for those tasks for which simulation-based training is indicated. The first subprocess of this activity (OPT/A2121) creates a scatterplot that shows the simulation requirement index and development cost index for each task. The second subprocess (OPT/A2122) summarizes the recommendations.

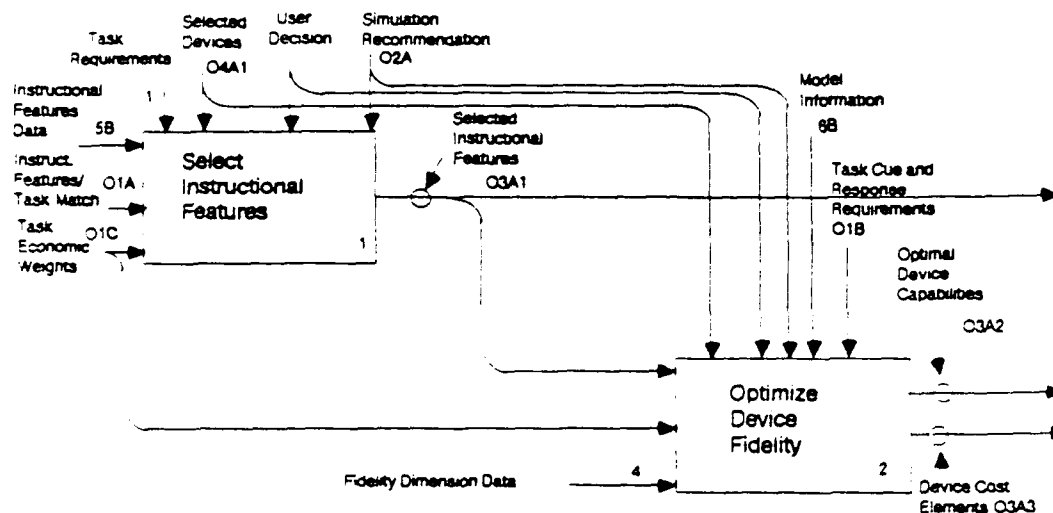


OPT/A3: Design Training Devices

Two tools are used to aid in the design of training devices. The first tool determines the cost-effective instructional features to be incorporated into a training device. The second tool determines the device areas in which technical sophistication and fidelity would be cost-effective.

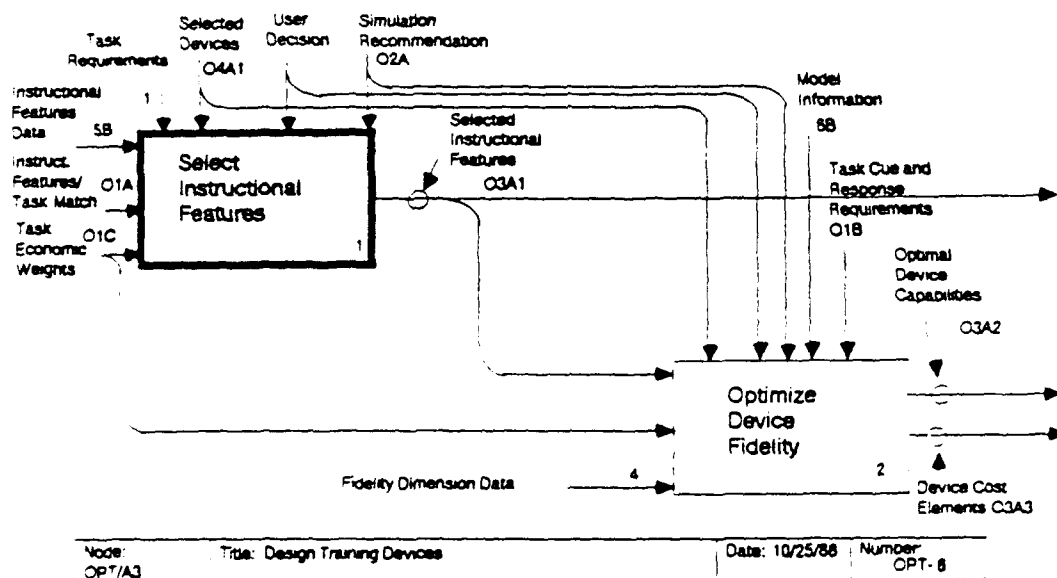
OPT/A31: Select instructional features. This tool examines the instructional features that would improve training efficiency for each task (O1A). Then the tool identifies the instructional features that have the greatest expected impact on training efficiency, given their cost. The output of this module is a list of the instructional features, ordered by decreasing benefit/cost ratio. Features at the beginning of the list provide a better value, given their development cost, than the features at the end of the list. The optimal set of instructional features at any budget may be determined by selecting features in order of decreasing benefit/cost ratio until the budget is met. The results of this model may be integrated into the analysis performed by the fidelity optimization model.

OPT/A32: Optimize device fidelity. This tool addresses the problem of how much should be invested in the technical sophistication and fidelity of a training device. The tool considers several dimensions of technical sophistication for simulating the real world that are related to task cue and response requirements. A



Node: OPT/A3	Title: Design Training Devices	Date: 10/25/88	Number: OPT- 6
-----------------	--------------------------------	----------------	-------------------

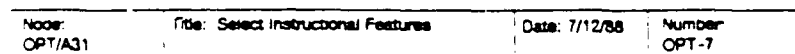
training device may be more or less sophisticated on each fidelity dimension. The module compares task requirements on these dimensions with the cost of meeting these requirements, in order to determine the areas in which an investment in increased fidelity can be justified by increased training effectiveness. The output of this model is a collection of training device designs that optimize training effectiveness as a function of investment cost. Given this set, the optimal training-device design at any development cost may be determined. The choice of the single training device design that meets training criteria at the lowest cost from the set of optimal device designs should be made by either the Training Device Selection Module or the Resource Allocation Module.



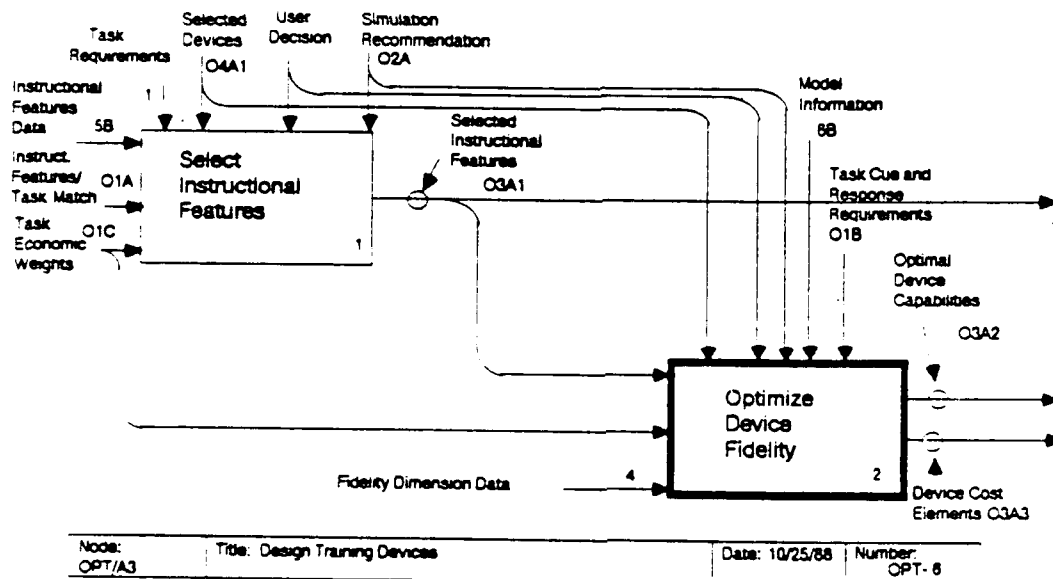
OPT/A31: Select Instructional Features

This tool for designing training devices identifies, given a development cost, the instructional features that will improve training for the most tasks. Instructional features are presumed to influence the efficiency of training rather than the maximum effect of training or transfer of training. Thus, instructional features influence the time multiplier in the learning model rather than the asymptote. Three major activities compose this tool: candidate features are selected from a master list, the benefits of these features are determined, and the features that provide the greatest benefit for the cost are determined. The outputs of this module may serve as inputs to the Fidelity Optimization Module.

OPT/A311: Select tasks and candidate features. In this activity, the user selects the training tasks to be used as the basis for the evaluation (O311B). This selection is described in node OPT/A3111. The selected tasks may be recommendations of the Simulation Configuration Module (O2A) or Training Device Selection Module (O4A), or the user may select the tasks on some other basis. In addition, the user selects the instructional features to be evaluated in the analysis (O311A) from a master list of instructional features (5B). This selection is described in node OPT/A3112. Specific procedures for this activity are described in the detailed diagram for this node.



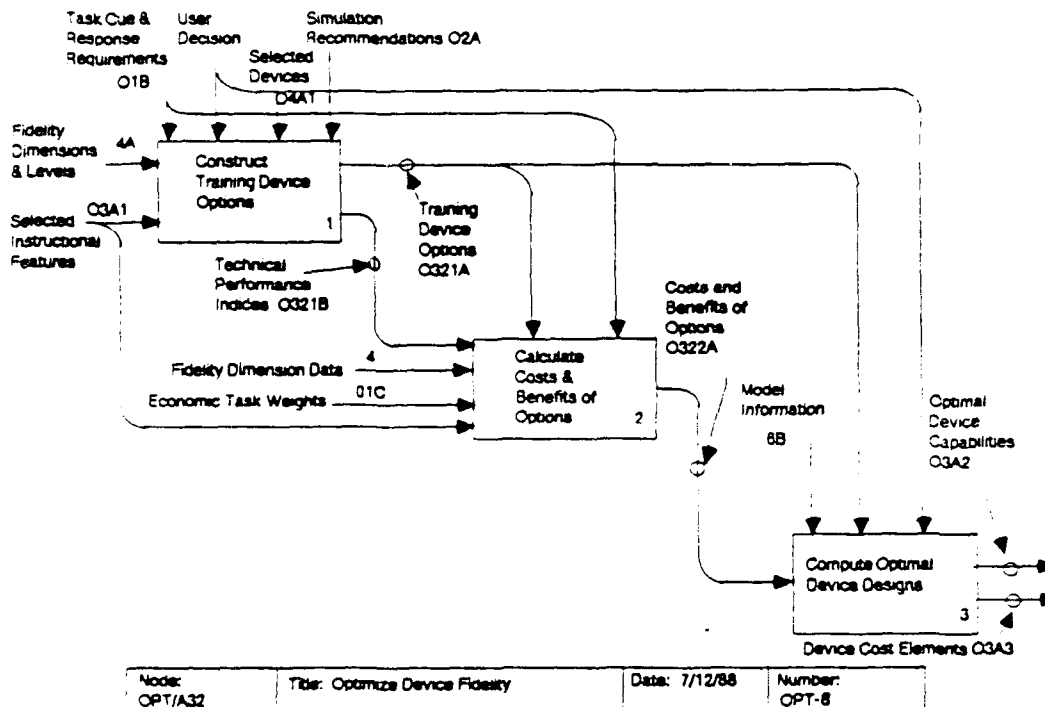
OPT/A313: Select optimal order of features. In this activity, the instructional-feature benefit measures (O312A) are combined with assessments of feature-investment cost (5B1), and the ratio of benefit to cost is used to determine the optimal selection of instructional features at any cost (O31A). The user may examine the optimal selection of features at any development cost. In addition, the entire ordered list of instructional features, or some portion of it, may serve as input to the fidelity optimization module. The optimal order of features is determined by computing the benefit/cost ratio for each feature (OPT/A3131), sorting the features by benefit/cost ratio (OPT/A3132), and listing the instructional features (OPT/A3133).



OPT/A32: Optimize Device Fidelity

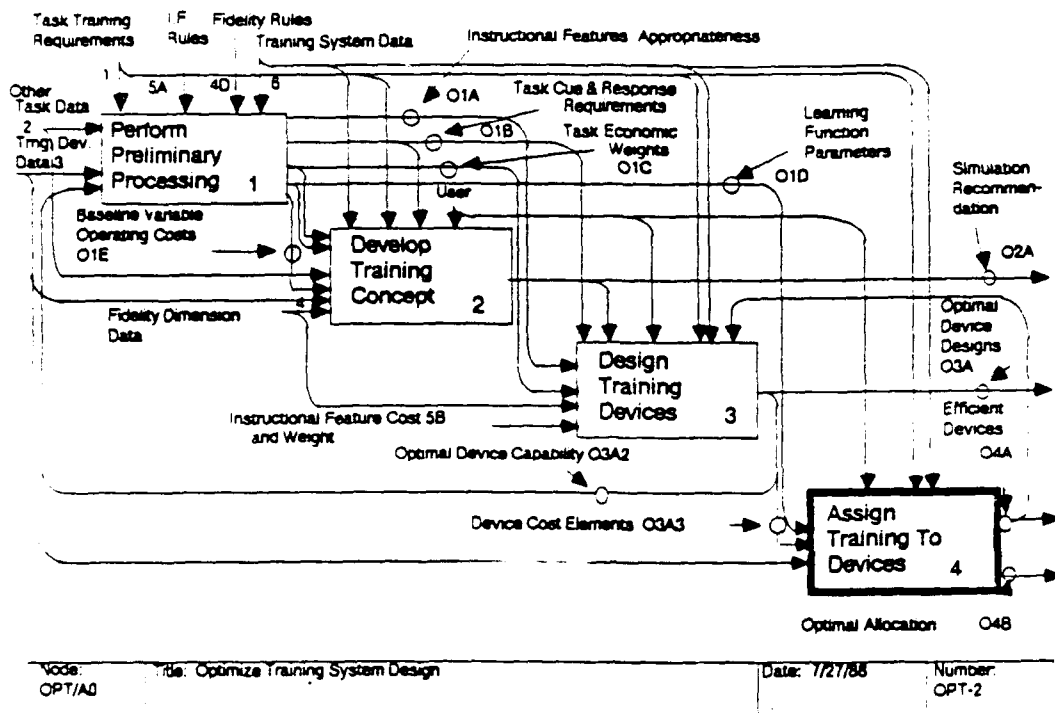
The goal of this tool is to consolidate task cue and response requirements for the tasks being considered for training on a training device with training device fidelity options so that optimal device designs can be identified.

OPT/A321: Construct training device options. This activity allows the user to examine the cue and response requirements (O1B) of the tasks specified in the training concept (O2A) to select relevant fidelity dimensions (in OPT/A3211) and pick the minimum and maximum options in each fidelity dimension (in OPT/A3212) from a master list of options in the data base (4A). The user is also asked to determine whether the results of the instructional features module, if available, should be incorporated as a fidelity dimension in this module (OPT/A3213). Finally the user may discard any options between the minimum and maximum for a given fidelity dimension (OPT/A3214). These options and dimensions have been designed to be as independent of each other as possible--independent in the sense that cost and benefit of each option do not depend on the options chosen for other dimensions. The outputs of this activity are a set of candidate training device options (O321A), and associated technical performance indices (O321B) from the data base (4A3).



OPT/A322: Calculate costs and benefits of options. This process has four operations that establish development costs and training benefits for each option. These costs and benefits are comparable across all fidelity dimensions and are based upon the technical performance indices associated with each option. Costs are determined from technical performance by a logarithmic estimation function in the first subactivity (OPT/A3221). The remaining three subactivities determine benefit by determining task trainability (OPT/A3222), calculating a benefit score within each fidelity dimension (OPT/A3223), and determining weights that place benefit on a common scale across fidelity dimensions (OPT/A3224).

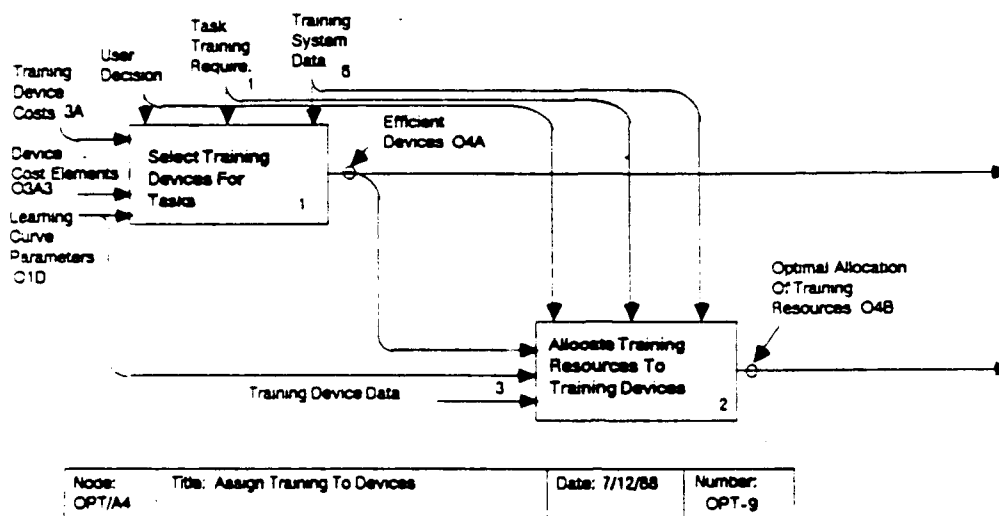
OPT/A323: Compute optimal device designs. This process has three operations. The optimal training device designs are based upon the incremental benefit-to-cost ratios of the options. After these ratios are computed (OPT/A3231) the options can be sorted in priority order (OPT/A3232), and optimal designs can be defined for user-specified cost or performance levels (OPT/A3233). Alternative optimal device designs may then be compared by the Training Device Selection module or Resource Allocation Module to determine which meet the training requirement at the lowest cost. The specific procedure for this activity is described in the detailed description for this node.



OPT/A4: Assign Training to Devices

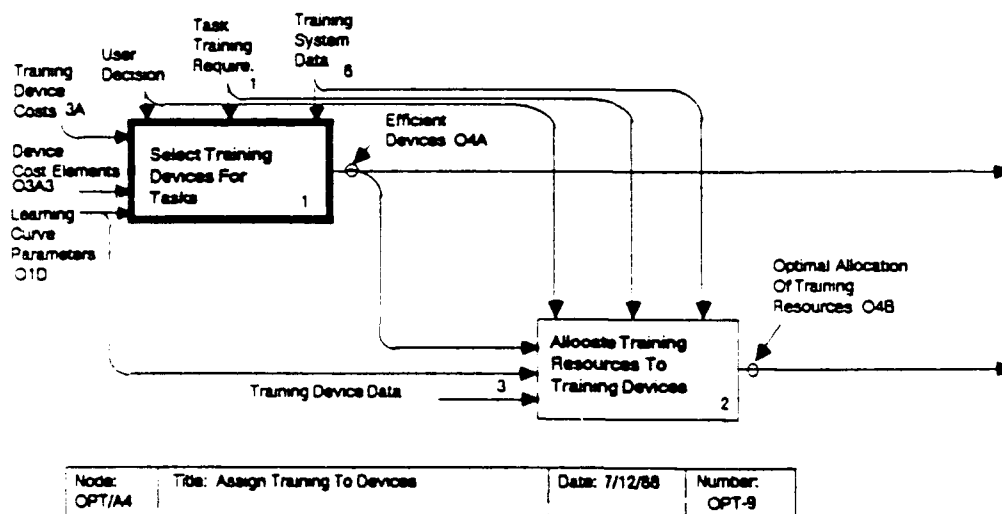
The goal of this activity is to assign training on each task among previously defined training devices. These training devices may have been designed by the modules in OPT/A3, or they may be templates of existing devices. In either case, the activity must determine the allocation scheme that meets the training requirements at the minimum cost. Two tools were designed to address this problem. The first tool makes simplifying assumptions to allow interactive operation. For example, this tool assumes that as many devices as needed may be acquired to meet training time demands. The second tool makes fewer assumptions, and permits user imposition of additional constraints on training devices usage. For example, it considers the discrete costs of device acquisition, and it allows the user to place a ceiling on the number of devices of each particular type that may be acquired.

OPT/A41: Select training devices for tasks. This tool, the Training Device Selection Module, determines a group of devices that should be used to train each task, and calculates the amount of time a student should spend training on each of these devices. The chosen training devices are conditionally optimal, in that they meet the training requirements at the minimum cost subject to the simplifying assumptions mentioned above. Those assumptions imply that training device cost



functions are simple linear functions of per-student training device use (time). This simplification allows the module to perform the optimization in seconds.

OPT/A42: Allocate training resources to training devices. This tool, the Resource Allocation Module, also determines device usage to minimize total training costs, but treats the cost relations in more detail and allows the imposition of certain constraints. In particular, it weakens the assumption of simple linear cost functions and allows for the inclusion of assets on hand. The new form for a training device cost function is assumed to be piecewise linear, with (possibly) an initial segment corresponding to assets on hand, and the subsequent segments corresponding to acquisition of another one of that device. The optimization conducted in this module considers constraints that limit the amount of time a training device may be used or the performance level at which the device may be used. As a result of the increased generality, the Resource Allocation Module will require significantly greater time for solution than the Training Device Selection Module. Consequently, its use may be more appropriate in a "batch" mode than in an interactive mode. Provision is made, however, to speed the algorithm by sacrificing some of the detailing in cost function representation.

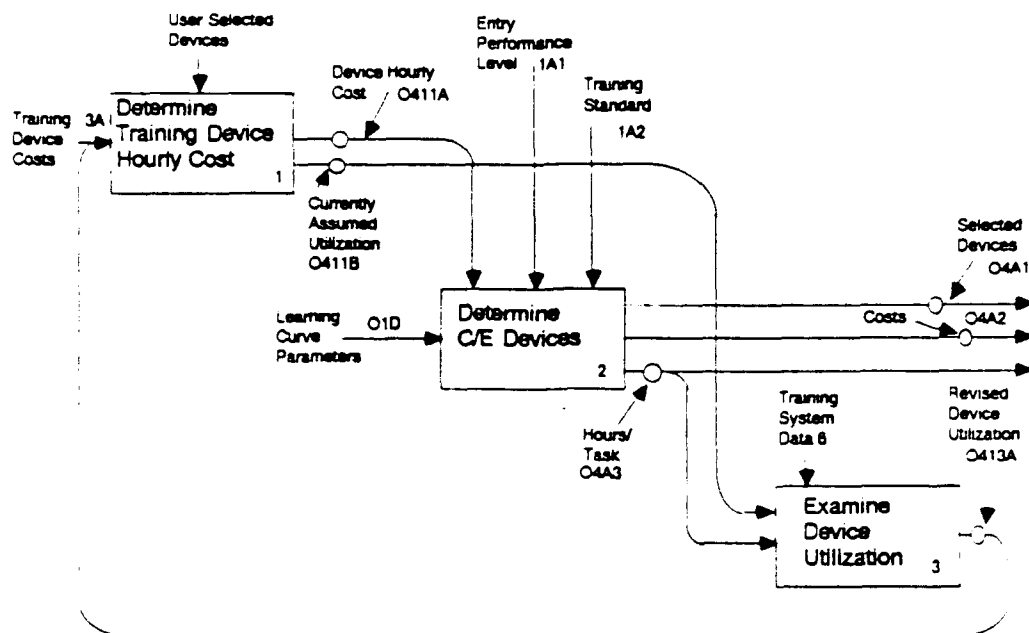


OPT/A41: Select Training Devices for Tasks

This tool applies cost-effectiveness analysis to select the training devices that meet training requirements at the minimum cost. The model considers (1) the fidelity of each candidate training device, (2) the instructional features present in each candidate training device that can reduce the time required to reach a training criterion, and (3) the level of training to be conducted on each training device. The model determines the devices that should be used to train each task, and the level of training at which a student should change from one device to the next device in sequence, in order to achieve required training standards at minimum total training cost. Model outputs include the associated training time required on each device.

OPT/A411: Determine training device hourly cost. This activity allocates the total life cycle cost of a training device over its expected number of hours use. The estimate is based on a nominal utilization ($UNOM_i$) in hours per year. In the initial run of the model $UNOM_i$ is set to the maximum annual utilization ($UMAX_i$) found in the data base (3A5) or calculated in OPT/A3234 (O3A3). On later iterations, the value of $UNOM_i$ that is updated in activity OPT/A4134 (O413A) is used in this activity. The total hourly cost for the device ($VTOTC_i$) is given by the following equation:

$$VTOTC_i = INV/(LC_i * UNOM_i) + FOC/UNOM_i + VHRC_i,$$



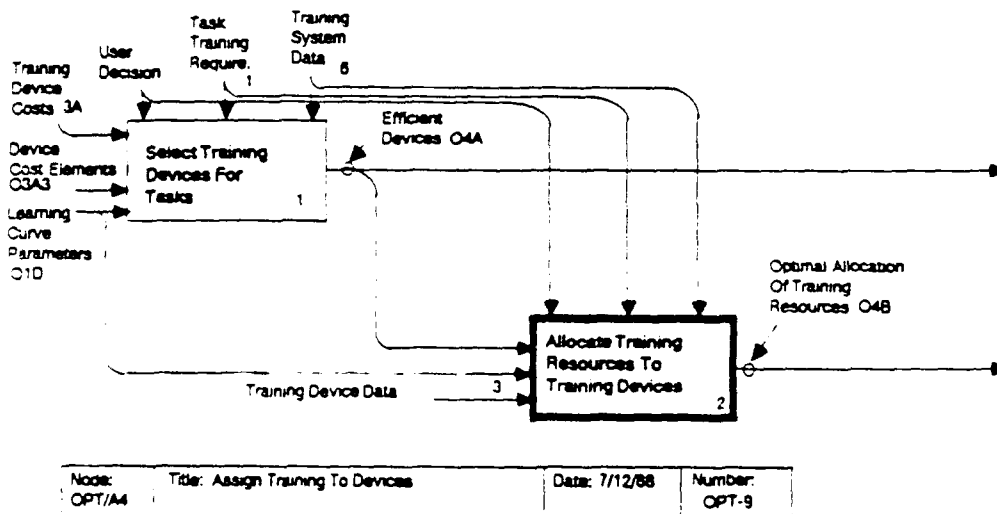
Node: OPT/A41	Title: Select Training Device For Tasks	Date: 8/1/88	Number: OPT-10
------------------	---	-----------------	-------------------

where the following parameters are taken from the data base:

INV_i = the investment cost per device for device i ,
 LC_i = the number of years in the life cycle for device i ,
 FOC_i = the annual fixed operating cost of device i , and
 $VHRC_i$ = the hourly variable operating cost of device i .

OPT/A412: Identify cost effective devices. This activity analyzes the relationship between each task and possible training device configurations (selection and sequence) to determine the training devices that can train from entry level to training standard at the lowest cost. It performs this analysis by choosing, for each performance level, that device providing the highest rate of increase in student performance as a function of the effective price of using that device. The specific procedure used is described in the detailed description of this activity

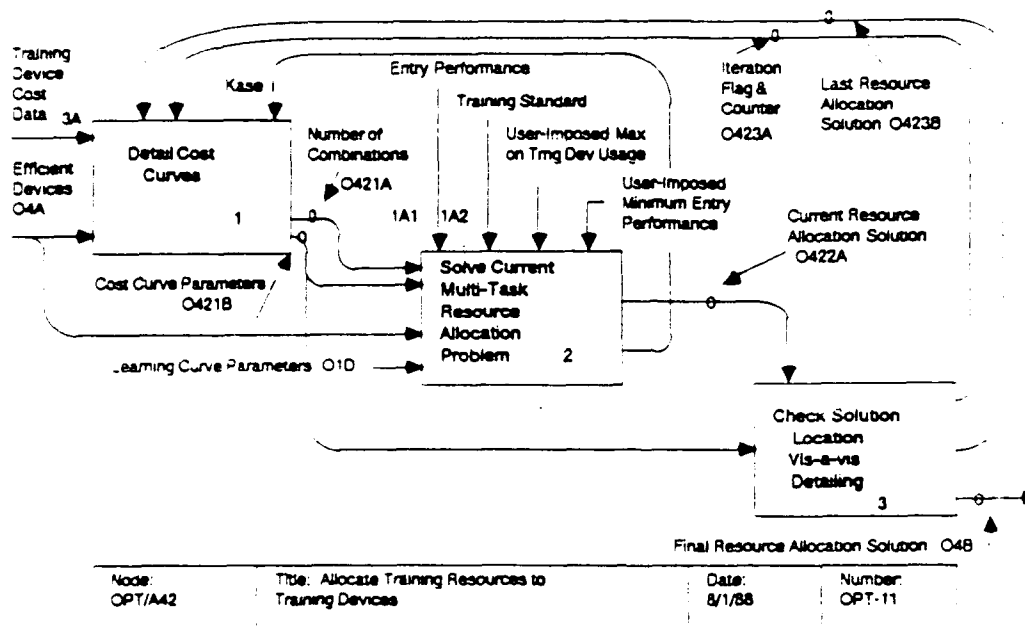
OPT/A413: Examine Device Utilization. This activity compares the value of device utilization that was assumed in the analysis ($UNOM_i$) with an actual utilization value, $UACT_i$, calculated at OPT/A4133. If $UNOM_i$ and $UACT_i$ are very different for device i , then that device may be considerably more or less efficient than was assumed by the analysis. In this case the user may decide to rerun the analysis adjusting $UNOM_i$ to have the value of $UACT_i$. The specific procedure is described in the detailed diagram for this node.



OPT/A42: Allocate Resources to Media

The resource allocation activity determines a refined training program minimizing total system cost to train students to criterion across tasks, subject to constraints on the amount of time a training device may be used or the performance level at which the device may be used. It employs a detailed cost function for each training device.

The algorithm to solve the resource allocation problem is heuristic and iterative. The general idea is to detail the device cost curves only around the solution. The strategy is to begin with relatively undetailed cost curves and generate a first solution using these curves. Then the cost curves are detailed around that solution and the cost minimizing solution is found anew. If the new solution is found to lie within the cost curve domains detailed, the process is terminated, and the solution is deemed optimal. If the current solution does not lie totally within current domains of detailing, the cost curves are redetailed around the current solution, and the cost minimization solution found. If the process is not terminated after a predetermined set of iterations, the process is terminated and the last solution found deemed "optimal", even though it lies outside the domain of detailing for one or more devices. In any case, the true cost functions are used to determine the per-student cost of training for the final solution.

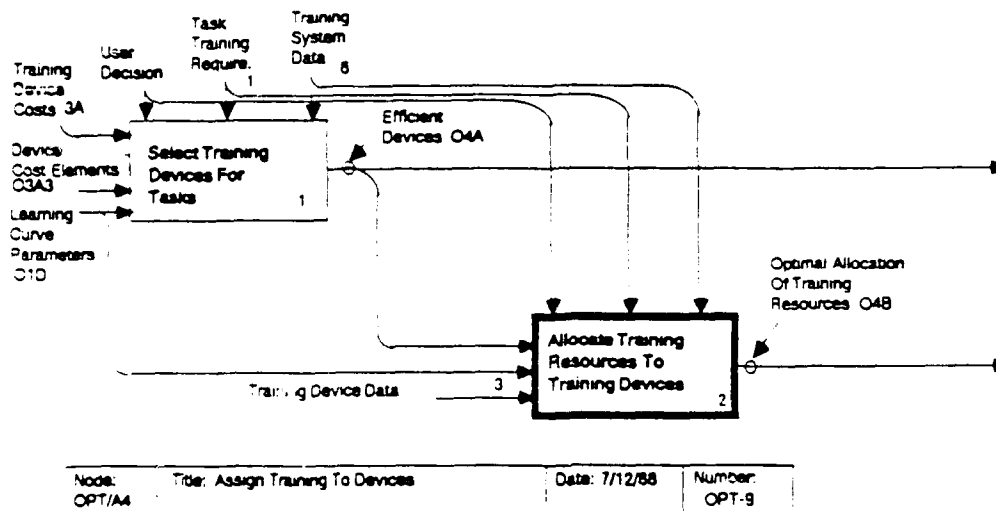


OPT/A421: Detail Cost Curves. On the first visit to this activity each cost curve is detailed into (at most) a two segment continuous, piecewise linear function. The first segment corresponds to employment of training device assets on hand, and the final segment corresponds to acquisition and employment of new assets. This simple two segment approximation is made to get the iterative solution process going.

On subsequent visits to this activity the cost curves are detailed for several segments on and around the per-student device usage found in the preceding solution. This will generally be a discontinuous piecewise linear curve--a staircase function with sloping steps.

On each visit to this activity the number of segments detailed must be limited to keep the total number of combinations reasonable, since the optimization problem will be solved for most, if not all, of the cost curve segment combinations. If the number of combinations is too large, then some of the curves are further simplified, at the expense of accuracy. The curves are selected for simplification based on least potential error introduced by the simplification.

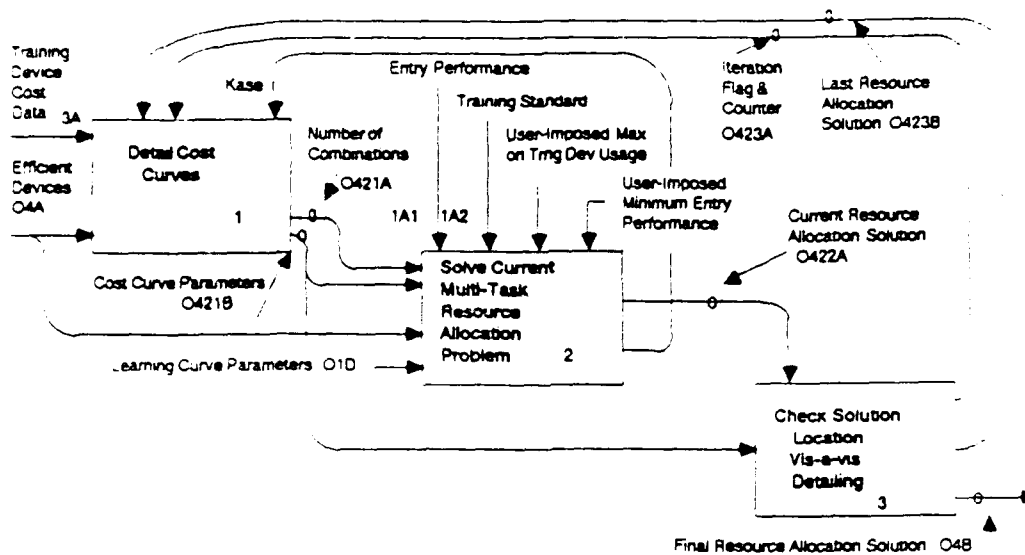
OPT/A422: Solve Current Multi-Task Resource Allocation Problem. On each visit to this activity the constrained multi-task resource allocation problem is solved under the current set of device cost functions detailed in OPT/A421. The



solution procedure involves solving the problem for each possible combination of cost curve segments.

OPT/A423: Check Solution Location Vis-a-Vis Detailing. The solution generated in OPT/A422 will specify a certain total per-student usage for each training device. If these usages fall within the domain of detailing for the respective device cost functions, then the solution has been found with proper consideration of cost curve detailing. If not, then the usage for one or more devices falls outside the domain of detailing, so will not represent desired accuracy. In that case, another iteration through OPT/A421 and OPT/A422 will be conducted, time permitting, in an effort to bring the solution into consonance with the domains of cost curve detailing.

If convergence is not obtained within a mandated number of iterations through OPT/A421 and OPT/A422, then the last solution obtained is used and the concomitant inaccuracy in cost curve detailing accepted. Even in this case the cost of the final solution is evaluated using the properly detailed device cost curves. Thus, failing convergence, the solution will still be feasible and accurately costed, but is expected to be somewhat more costly than a truly optimal solution. Experience to date suggests that the optimization error will tend to be small even if convergence is not achieved.



Node: OPT/A42	Title: Allocate Training Resources to Training Devices	Date: 8/1/88	Number OPT-11
------------------	---	-----------------	------------------

OSBATS Model Implementation

The primary goal of the second phase of the project was to develop software that embodies the OSBATS model. The resulting software provides a prototype decision support system (DSS) that can interact with an analyst about an existing training system in Army Aviation. The specific example is based on the AH-1 Airman Qualification Course (AQC).

The OSBATS software runs on an IBM PC/AT, Zenith 248 or compatible with 640K of memory, and a 10 megabyte hard disk. In addition, the following features are required: (a) an enhanced graphics adapter (EGA) and color EGA monitor, (b) 80287 numeric coprocessor, and (c) a Microsoft-compatible mouse. The software was developed using the Microsoft C Compiler (version 4.0) and the EXSYS expert system shell (version 3.2).

Individual modules were delivered for evaluation as they were developed. We used the preliminary evaluation results to guide the development of later modules. Modules took between one and two months to develop. We delivered the integrated OSBATS prototype (version 1.0) approximately one year after the completion of the model descriptions. We then revised Version 1.0 to add new functions, provide for additional analyses, fix detected problems, and increase the extent of integration. When the revised version was evaluated, the results suggested some additional minor revisions, which were made. The final version is denoted as version 1.1 with a date of 31 October 1988.

The software implements all of the modeling capabilities described in the model design. The software allows the user to define task clusters, develop candidate training-device designs for individual task clusters, and evaluate the cost of providing training using these designs in various combination with or without actual equipment. The software does not provide the capability to enter, or modify the data required by the model. That capability is being investigated and prototyped in a separate effort (Willis, 1988).

In the normal DSS development process, the problem must be represented to the decision maker in a way that is consistent with his or her internal representations of the problem. In addition, the DSS must supply the user with the capability to switch between representations, alter the values of input data, perform ancillary analyses, and otherwise control the operation of the DSS. An important component of system analysis for DSS development is the specification of representations, operations, memory aids, and controls used by the system. This kind of analysis has been described by Sprague and Carlson (1982), who term the methods the ROMC approach. Use of a process-independent system analysis, such as the ROMC procedure, provides for a problem-centered user interface. This interface provides the information to the user in a natural format, and hence enhances the ease of use and ultimate usefulness of the DSS.

An ROMC analysis is often based on interviews with the intended user of the DSS. In these interviews, the user is asked to explain the problem and the methods used to solve them. Representations are inferred from the constructs used to explain the problem. For example, if the user explains the results using a graph or a table, then that graph or table is a representation that the DSS should support. The other components are identified by similar analysis of the user's description, and by considering the requirements of the problem.

In OSBATS, because the analytical development of the DSS provides a considerable change from existing methods, we expected that use of OSBATS would encourage new representations of the problem domain that the user could not identify before the DSS was developed. Consequently, the process-independent analysis was generated by contractor staff analysts who had used methods similar to those specified in OSBATS in different contexts. The representations, operations, memory aids, and controls identified in the analysis were used as the basis of storyboards that described the DSS from an operational viewpoint. The storyboards, in turn, were used to develop the DSS modules.

This section gives an overview of the overall OSBATS software capabilities and features. First, it describes some of the general features of the software. These general features include some of the general control functions provided by the software. Then, it summarizes the displays generated by the software. The displays provide both the representations and the memory aids supported by the software. Finally, the section describes the operations that the user may perform to affect the model results.

General Features of the Software

The user-interface consists of a variety of menus and displays. The user primarily interacts with the software using the mouse. The various modules are accessed through the Main Module Menu. Likewise, each module has a menu of its own outlining its subsections. Each module subsection contains a display or series of displays. A display may be a graph or a table of results or it may be a display designed to obtain user input. The user can access various displays within a module subsection through a menu at the bottom of each display. OSBATS also includes a "help" feature to assist the user, which can be accessed from any display, and simply describes the screen that was active when help was selected. The figures presented in this section depict the displays produced by OSBATS, except that the control menus and status header are omitted.

The general features described in this section are available to the user in all modules and provide methods for the user to control the operation of the software. General control is provided by the following five features.

1. Help. A help option is presented at most displays. The help screen describes the information that is included in the display and lists the options that are available from that display.
2. Print screen. Any display screen can be printed on a standard printer. The printout includes model status information as well as the information contained in the display, but excludes menus. Both tabular and graphical displays may be printed.
3. User comment. The User Comment feature is available for recording notes and comments as OSBATS is running. These comments are saved in files for later review.
4. Model trace. The OSBATS software automatically keeps a record of the most recent run. The record contains a list of each screen that was viewed and the number of the comment(s) made within that screen.
5. Screen comparison. For selected displays, the user has the capability to save the current content of displays for later examination within the current run of the model. This feature may be used to compare results under different assumptions.

These general features are supplemented by module-specific features. The module-specific features are summarized in the following sections. All model features are described in detail in the User Guide (Gilligan, Elder, and Sticha, 1988).

Summary of Module Displays

Module displays are designed to provide both representations of the model results and memory aids that list some of the model's assumptions. The display designs considers two needs that must be satisfied by the displays. First, the displays must portray both general and detailed results. The general results present an overview to the user in a single display. The detailed results present justification for the overall results. Without justification, there would be no way for the user to develop any confidence in the model's recommendations. Because of this need, we designed displays at several levels of generality for each module. For example, the most general display might summarize the results of an analysis that aggregates benefit and cost. A somewhat more specific display would show individual benefit or cost values. A still more detailed display would show the task variables that were the basis of the benefit values.

The second requirement considered in the display design is the need to present information in different formats. This need was addressed by designing complementary displays that had either tabular or graphical formats. Often, the tabular and graphical displays contained the same information, but because of the different characteristics of the display formats, results would be highlighted differently. In other instances, the different display formats contained somewhat different information, although they were at the same level of detail.

All screens used to display analysis results have a common appearance. The screen is divided vertically into three sections. The top lines give model status information, such as the current task cluster, and the status of weights used in the analysis. The bottom lines give a menu of options available to the user. The middle section of the screen presents the table, graph, or list describing the results.

The representations and operations for the OSBATS software are illustrated for the Fidelity Optimization Module. We implemented four representations in the prototype. The first, or matrix representation (Table 2), describes the options that are available for evaluation in the model. A particular system design is determined by choosing one cell from each row in the matrix. The matrix may display several different kinds of data, including costs, benefits, and benefit/cost ratios. The second, or graph representation (Figure 7), shows the cumulative benefit and cost of those system designs that provide the greatest benefit for the cost. The third, or package representation (Table 3), describes a particular optimal system design. The fourth, or user-defined package representation uses the same format as the package representation. However, in this representation, the user may define a design, which may or may not be optimal. The system analyzes the users' design to determine the overall cost and benefit. The users' design may then be compared to optimal system designs that have a similar cost or benefit.

The representations are linked so that they give consistent views of the problem. Thus, the design associated with a point identified on the graph may be viewed in either the matrix or package representations. Several operations are possible within these representations. These operations include, selecting the data to view on the matrix representation, finding more or less expensive optimal options, finding the optimal design at a criterion cost, and eliminating options from consideration by the model. In addition, the user has control of the model that operates on all representations. The user may define the requirements for the training device being designed using the model, and may change aspects of how the benefit values are calculated.

Table 2. Matrix Representation Showing Incremental Benefit-to-Cost Ratios for Fidelity Levels

Level	Benefit/Cost Ratio						
	1	2	3	4	5	6	7
Visual_Resol.	undef	----	----	----	0.12	----	
Visual_Content	undef	0.88	0.01	----	----	----	
Visual_Texture	undef	----	----	1.36	----		
Visual_Front	undef	2.49	0.13				
Visual_Side	undef	----	----	----	1.64	----	0.07
Point_Effects	undef	----	----	----	----	2.76	
Area_Effects	undef	4.51	----				
Platform_Mot.	undef	----	0.00	----			
Seat_Motion	undef	11.16	0.48				
Sound_Effects	undef	----	11.13	0.95			
Map_Size	undef	0.66	----	----	0.00	----	----
inst_features	undef	0.78	0.54	0.40			

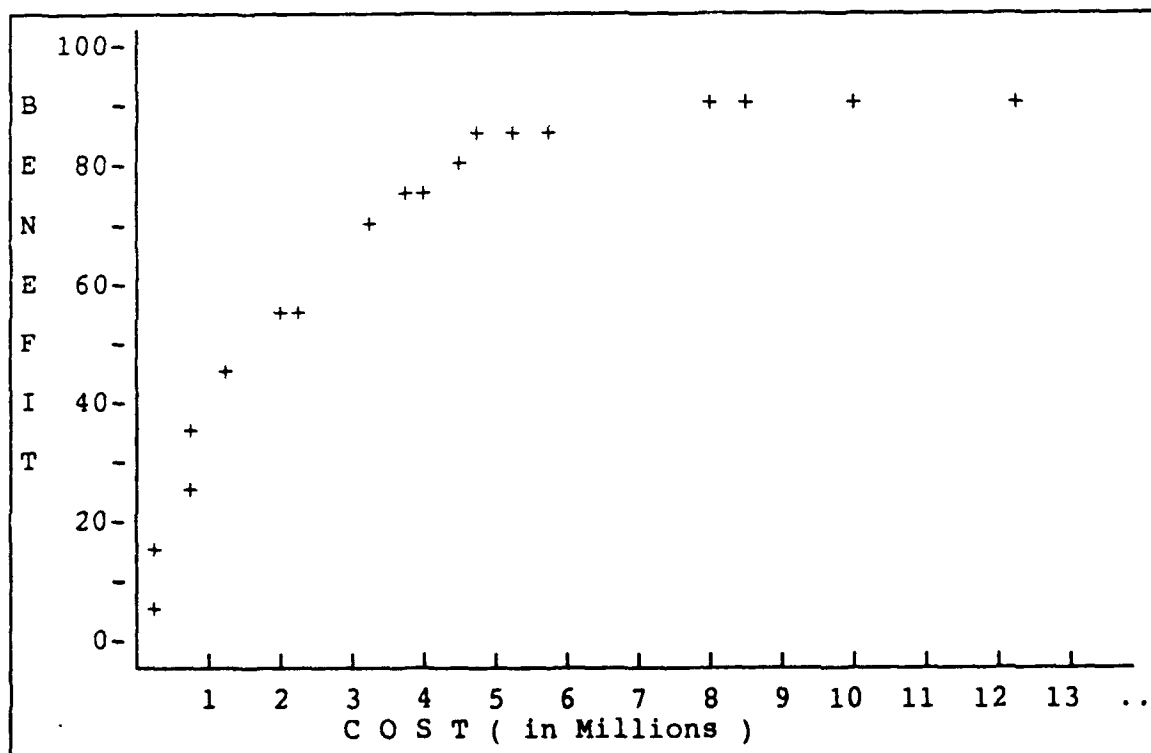


Figure 7. Graphical representation of Fidelity Optimization Module results.

Table 3. Package Representation Showing Optimal Training-Device Design at a Cost of \$5.5 Million

Dimension	Level	Description	Benefit	Cost
Visual_Resol.	1	m2 at 0.3 km	0.0	50.00
Visual_Content	2	Add Generic Featrs	3.7	496.08
Visual_Texture	4	More Digit Ph	14.4	1108.32
Visual_Front	3	40 x 60 degrees	10.5	1103.93
Visual_Side	5	40 x 50 degrees	4.9	194.67
Point_Effects	6	Add mvng Gnd Veh	15.7	504.00
Area_Effects	2	smoke and dust	7.8	147.22
Platform_Mot.	1	none	0.8	0.00
Seat_Motion	3	Add G-Seat	8.2	192.00
Sound_Effects	4	Add abnor opt nse	11.6	192.00
Map_Size	2	10 x 10 km	3.6	634.40
Inst_features	4	Inst Feat, Lev 4	6.0	874.00
Total			87.1	5496.62

Summary of User Functions

The user has the following options that can affect the recommendations of the model at different points of the analysis.

1. Task cluster definition. The initial definition of task clusters is provided by the Simulation Configuration Module. However, the user may modify the clusters defined by this module, or may define task clusters independent of the recommendations of the Simulation Configuration Module. These clusters may be used as the basis for the analysis performed by the other modules, except for the Resource Allocation Module, which is based on the entire set of tasks.
2. Inclusion of weighting factors. Several of the modules include weights for tasks, instructional features, fidelity dimensions, and so forth. These weights represent the relative importance of factors that are included in the analysis. In some cases, the user may set the value of the weights directly. In other cases, the user may choose either to use a set of weights calculated by the model or to set the weights for all tasks, features, dimensions, or other factors.
3. Selection of range of device options. The maximum range of fidelity options considered by the Fidelity Optimization Module is specified in the resident data used by the model. The user may restrict this range at any time to reflect budgetary limitations, direction from higher authority, or other factors.

4. Selection of devices from recommended alternatives. The Instructional Feature Selection and Fidelity Optimization Modules recommend a range of optimal designs that vary with respect to their cost. The user may select designs from the list of optimal designs for evaluation by later modules.
5. Construction of device family. Ultimately, a training system consists of a family of training devices. After the user has determined several alternative training device designs, he or she may combine these designs in any way to produce an overall approach to meeting the training requirements. The software will allow the user to evaluate the alternative training concepts.
6. Specification of device use constraints. Use of training devices or actual equipment may be restricted for several reasons. The user may set limits on either the overall use of a training device or actual equipment or the minimum performance level at which the equipment may be used. The Resource Allocation Module then allocates training time to the devices subject to the constraints.

The number and importance of the user functions indicate that the user can have a major impact on the results of the model. The model software provides a comment function as a means for documenting the assumptions and user changes that form the basis of the recommendations.

Evaluation of the OSBATS Model

Formative evaluation is a critical component of the development of the OSBATS DSS. We conducted evaluations of the model and software at several points in the development process. The evaluation effort had the following goals:

1. To determine the extent to which the OSBATS model is relevant to the training-device design problems actually faced by engineers at PM TRADE,
2. To determine the extent to which the data required by the OSBATS model are currently available, or could be made available with reasonable effort,
3. To verify that the OSBATS software accurately performs the functions specified in the model documentation,
4. To determine the ease of using the system and comprehending its outputs,
5. To determine the validity of the predictions made by the system,
6. To determine the generality of the OSBATS procedures and data to different training domains.

Our evaluation effort consisted of three types of activities. The first type of activity involved interviews with potential model users to determine the relevance of the model, the data requirements, and the ease of use of the software. These interviews were first conducted when the individual OSBATS modules were developed. The initial interviews were quite informal, and were combined with a briefing of the function of the module. The results of the initial interviews were used to guide the development of later modules. After all modules had been developed and integrated, we conducted more extensive and formal interviews with engineers from PM TRADE. These interviews evaluated all modules in the OSBATS system. Some of the results of these interviews were incorporated into version 1.1 of the OSBATS software. Other results provide the basis for the recommendations for model and software enhancements discussed in the "Model Description, Implementation, and Evaluation" report (Sticha, et. al., 1989).

The second type of evaluation activity involved using the model on problems designed specifically to illuminate aspects of model performance. The accuracy with which the software represents the model was verified by comparing the results of the software to selected problems with results calculated by hand. Where discrepancies were found, we determined the source of the discrepancy, and communicated our results to the software-development staff. Our verification of OSBATS version 1.0 found

several discrepancies. Our evaluation of OSBATS version 1.1 verified that these errors were corrected. We are now confident that the calculations used in the software correspond to the current model documentation.

The performance of the OSBATS model and, to some small extent, its validity, were investigated by conducting several sensitivity analyses. These analyses varied model inputs and allowed inspection of the resulting recommendations. The analyses were conducted to ascertain whether changes in input values had expected results, and to discover which inputs variables were the most important in determining the model recommendations. The results of the sensitivity analyses were used to guide the revised research plan and are also summarized in this section.

The third type of evaluation activity investigated the generality of the model to new training domains by analytically applying the OSBATS model (not the DSS) to armor maintenance training. This analysis investigated the extent to which data variables and modeling constructs would need to be changed to apply OSBATS in this new domain. The details of that analysis are also reported in this section.

Structured Interviews of Potential Users

To obtain input from potential OSBATS users, we interviewed five PM TRADE engineers. Each interview was structured to include a directed demonstration of OSBATS and a set of questions about its operation. The interview addressed each of the five OSBATS modules, and each engineer evaluated (a) the presentation of data, (b) the clarity of the module's results, (c) the validity of the module's approach, (d) the availability of the data required by the module, and (e) the degree to which all relevant information was included in the module. As a group, these engineers have been involved in training-device development for an average of 12 years, with a range of 4 to 20 years. Their average time with PM TRADE has been 10 years, with a range of 2 to 20 years.

Many of the comments made by the engineers represent general areas of concern that should be considered in further OSBATS development. These general areas include the following:

1. OSBATS does not accommodate school requirements as constraints in the training-device development process. For instance, in the Instructional Features module, several engineers mentioned that the school often specifies certain requirements. Levels of instructional feature packages might be more beneficial if the user could enter into the system the features that are specified by the school and must be included in the base package.

2. A major concern is the cost data and the cost-savings associated with simulators. Engineers like cost comparison charts. Several engineers indicated that training-device design process is driven by cost, time schedules, and school requirements, which needed to be better reflected in OSBATS.
3. Development time seems to be an important issue that is not addressed in OSBATS. Time schedules are often constraints in the current training-device development process.
4. Engineers are confused by the derivation of the benefits of both instructional features and fidelity dimension levels. More explanation is necessary.
5. There is some confusion about how the results are normalized. Normalized numbers are confused with percentages.
6. "Functional groups" of tasks are often used in the current training-device design process (i.e., tasks requiring motion systems). There seems to be an inclination to look for "grouping" and to design device components according to the needs of the functional group.
7. Several engineers indicated that training-device design usually involves concentrating on one device as opposed to a "minimum family of devices." One engineer said, "usually the school won't consider families." He said later, "Our requirements are in terms of a device, not a combination of devices." Another engineer said that he usually tries to come up with "a single design."
8. There was concern with the availability of the data required by OSBATS and the cost involved with having to gather data for each different problem addressed. One engineer asked, "How usable is the system?" His opinion is that "in its present form, not being flexible and needing lots of data," the system is limited in use. Another engineer said that it is important that the user be able to get into the system and "inspect or change data on benefits and cost."

These general comments can provide guidance for future development of the OSBATS model. Both model and software revisions will be required to address these issues. In addition to these general comments, the engineers had a number of specific comments about aspects of the operation of the software, adequacy of displays, use of color, user interaction, and so forth. Some of suggested changes have already been incorporated into the OSBATS software. Others are beyond the scope of this development effort, and must be addressed at a later date.

Sensitivity Analyses

Sensitivity analysis refers to a set of procedures used to assess the effects of manipulations of a model's variables on the results of the model. These techniques rely on making systematic changes in the values of input data and measuring the resulting changes in model results. The sensitivity analyses had the following two goals.

1. To determine the responsiveness of the model to variable manipulations, and
2. To ascertain the validity of critical model assumptions.

Because sensitivity analyses do not involve empirical human performance data, they do not provide a rigorous test of the model validity. Rather, they test validity by ascertaining the correspondence of model predictions to general trends and model developer expectations. We describe the results in relationship to these two goals.

Model Responsiveness

We addressed model responsiveness by varying the major model inputs over a wide range of values and determining the effect on the model recommendations. In most cases, we used the results of the Training Device Selection Module as the main dependent measure to assess sensitivity.

Task training requirement. We investigated the effects of changes in entry performance level, performance standard, and task cue and response requirements. We found that lowering the entry performance causes the less expensive devices to be used in the initial training of a task. However, the performance level at which training switches from the less expensive device to the more expensive device (the crossover point) remains the same. The entry performance affects the training times through the head start and the time scaling factor. Increasing the performance standard causes a roughly proportional increase in the crossover points. Thus, when the performance standard is increased, the same training devices are generally recommended.

Task cue and response requirements are among the most critical variables in determining the recommendations of the OSBATS model. Increasing the cue and response requirements causes a decrease in the estimated device training effectiveness. Such a decrease in training effectiveness causes a shift in the training allocation away from low-cost devices toward the more sophisticated training devices and actual equipment.

Other task data. The major variables that were investigated here involve estimates of training hours and costs. We found that multiplying the task training hours by a constant factor produces a change in training times by exactly that factor. The relative proportion of training time allocated to each device

does not change, however. This result indicates that the devices chosen by the Training Device Selection Module are completely insensitive to the task training hours, when all time variables are varied in proportion. Changes in training hours would have an impact on the results of the Instructional Feature Selection and Fidelity Optimization Modules, however, because they would affect the task weights.

Training device data. The training device cue and response capability has a large effect on the model recommendations that is analogous to the task cue and response requirements. The effects are not completely analogous, however, because the effect of the cue and response capability is moderated by the minimum performance parameter. We found that the change in cue and response capability has the largest influence on the asymptotes when the minimum performance parameter is low, and hence, the fidelity dimension is more critical.

Fidelity dimension data. We examined two fidelity dimension data variables, fidelity dimension cost and the fidelity dimension minimum performance parameters. We found that as the exponent in the fidelity dimension cost estimating function gets very large (ten times its original value), the cost curve becomes increasingly less linear. Low levels of technology are still cheap. Higher levels of technology get dramatically more expensive.

Lowering the minimum performance parameters for all fidelity dimensions to zero causes a decrease in training on the relatively unsophisticated devices. Increasing the minimum performance parameter for all eleven dimensions to 90 percent of maximum value causes a shift in allocation of training towards the less sophisticated trainers and away from the full-mission simulators.

Training system data. We examined three training system variables, the maximum instructional feature effect, the assumed setup savings percentage and the maximum number of instructional features. None of these variables had a very dramatic effect on the recommendations of the model.

Implications of Analyses on Validity

Several of the results of the sensitivity analyses help confirm the face validity of the OSBATS model, in that they correspond to our expectations. Other results offer important characterizations of the model that would provide the basis for a critical test of model assumptions. The following are some of the more important results, restated in somewhat more general terms.

1. Simpler training devices are more appropriate at lower skill levels: higher fidelity training devices are more appropriate at higher skill levels.

2. The most critical process in the model application is the process that determines task cue and response requirements.
3. The model predicts that fidelity can be sacrificed more readily on less critical fidelity dimensions than it can on more critical fidelity dimensions.
4. Fidelity is more important than instructional features at high skill levels; instructional features are more important at low skill levels, although it is difficult to say at what skill level instructional features become more important than fidelity.
5. Fidelity requirements are related to the performance standard. If the standard is raised, then it may require greater fidelity to train to that standard.
6. The total training hours required to train a task do not effect which training devices are selected for that task, but it does affect the total cost of training.

Most of these results are expected, and serve to give us some confidence about the validity of the OSBATS model, albeit on a very informal level. Other results, particularly those that relate the importance of fidelity and instructional features and tie fidelity requirements to the performance standard, provide the opportunity for critical tests of the OSBATS model.

Tank Turret Mechanics Analysis

The OSBATS model was created to be a general tool to aid the training device concept formulation process. However, its only application has been to a sample problem from the domain of aviation. Application to other areas may require changes to data values, model parameters, variables, fidelity dimensions and levels, instructional features, or even to the overall model structure. This section describes an analysis of the nature and extent of changes required to apply the OSBATS model to a different domain, specifically that of armor turret maintenance.

The specific domain analyzed is the M1 Abrams Tank Turret Mechanic course (45E10). The proponent for this course is the U.S. Army Ordnance Center and School at Aberdeen Proving Ground, Maryland. However, the course is taught at the U.S. Army Armor School at Ft. Knox, Kentucky. This course differs in many important respects from the AH-1 Airman Qualification Course (AQC) that formed the basis of the initial application of the OSBATS model.

1. The turret mechanic course involves maintenance, while the AH-1 AQC involves operation.

2. The turret mechanic course involves initial acquisition of skills, while the AH-1 AQC involves transition of skills to a new weapon system.
3. The turret mechanic tasks are heavily loaded on procedural and cognitive activities, while the AH-1 tasks are heavily loaded on psychomotor activities.
4. The turret mechanic course is taught to enlisted personnel immediately following basic training, while the AH-1 AQC is taught to officers and warrant officers.

The great differences between these two courses provides a good examination of the generality of the OSBATS model. In our analysis of the turret mechanic course, we concentrated on two questions. The first question is whether the OSBATS process will work in a training domain that is considerably different from its original application. The second question is what changes will be required to the data and procedures for successful application of the OSBATS model in the new situation.

One focus of our comparison of the two training domains was on whether we can develop general procedures to specify the appropriate data variables, such as fidelity dimensions and instructional features; determine measurement scales; and assess data values for a new training domain. The extent to which those procedures generalize will have a great impact on the operating procedures used by the OSBATS model and will to a great extent determine its general applicability.

Our analysis did not uncover any changes in the general OSBATS model process that would be required to make it applicable for optimization of armor maintenance training systems. However, the analysis indicated several differences in the two domains that could have implications on the operation of the model.

The first difference is in the complexity. The aviation training example involved more tasks, more fidelity dimensions, and greater variety in the skills trained. This apparent difference is partly illusory, however. The turret mechanic must be able to perform a far greater number of tasks than are covered in the school's Program of Instruction. Many of these are listed as "related tasks," and are not trained under the assumption that the ability to perform these tasks will transfer from other tasks. If the related tasks were included in the analysis, then the OSBATS model would need to be changed, because the possibility of transfer between tasks is not currently considered in the model.

A second difference is in the methods that must be used to determine fidelity requirements. The analysis of visual fidelity requirements in the fidelity rule base for aviation involves a detailed analysis of the kinds of activities required to perform a task, e.g. such specific actions such as estimating altitude or

range, or detecting distant targets. It seems that the analysis for the turret mechanic would not be as complex, and would not require the same depth of knowledge about the task. Thus, it is more likely that the engineer using the OSBATS model would be able to provide the data for the fidelity rule base in maintenance problem, while considerably greater subject-matter knowledge would be required to provide comparable data for aviation operations.

A third difference is that the reasons for simulation are somewhat different in the two domains. This difference would have an impact on the kinds of factors that are considered in evaluating the benefits that may be derived from device-based training.

The overall conclusion of the analysis is that the OSBATS model is applicable to the M-1 Abrams Turret Mechanic Course. No serious changes would be required in the general model processes and organization of modules. However, application of OSBATS to the new domain would require considerable development of resident data, particularly fidelity dimension data and rules. It must be noted that this effort was analytical in nature, was conducted by personnel who were very familiar with the OSBATS model, and did not encompass any empirical application or data development for the model.

The most difficult part of the modeling process is in the specification of the fidelity dimensions and levels. The complexity of the model is a function of both the system being used by the students and the environment in which the students use that system. For the flight trainer the system was the helicopter, which is considerably more complex than the test equipment of the maintenance trainer. However, the environment of the maintenance trainer was a complex tank turret, in some ways as complex as the aviation training device. The process of breaking the system and environment into dimensions and then defining levels of fidelity for each remains an art, but the process is much better understood now that it has been completed twice and should be codifiable in the near future. What is required is feedback from maintenance trainers on the dimensions and levels described above and the development of dimensions and levels for at least one more application.

Once the fidelity dimensions and levels are specified, the model process proceeds systematically. The OSBATS data base must be developed around the definitions of these dimensions and levels. All of the data elements are defined, although the data collection procedures are not codified; most of the data must be developed with the extensive support of SMEs.

Summary and Conclusions

The high cost of training using actual weapon systems and the expanded capability of training technology have increased the potential value of simulation-based training. However, the complexity of weapon systems and their associated training systems has made the process of designing training devices much more difficult. The process of formulating a cost-effective training-device concept requires many tradeoff analyses that compare the cost and effectiveness of alternative design concepts.

The OSBATS model is intended to aid the person (e.g. engineer) responsible for formulating a training-device concept. The goal of the OSBATS development has been to provide methods to produce training device designs that meet the training requirements at the minimum cost. The OSBATS system provides prototype tools to aid the tradeoff analyses required to design cost-effective training devices. The model begins to allow design engineers to consider training effectiveness seriously when they develop a training-device design concept. It attempts to provide an interactive environment that allows the user to consider many more alternative designs than would be possible without the model. Using the OSBATS model, the user can begin to perform comparative analyses that identify cost drivers, produce and evaluate alternative training-device design concepts, and specify cost-efficient ways to use training devices to meet the training requirements.

In this section of the report, the accomplishments made during this effort are described and the knowledge gaps that need to be addressed by future research are identified. The summary focuses on two issues. First, we highlight the major accomplishments of the development effort. Second, we summarize the activities that are required for validation and technology transfer.

Significant Accomplishments of this Research

Our three-year effort to develop the OSBATS model has produced several advancements in the state of the art for training-device optimization. These advancements build on the results of previous research and existing models. The following paragraphs summarize the most significant accomplishments, which distinguish the OSBATS model from predecessor models.

First, the OSBATS system provides a consistent approach for addressing a variety of training-device design problems. Its' consistency comes from the top-down design and the coordinated use of cost-benefit optimization in each component. Each module addresses one aspect of the training-system design process and recommends an optimal choice by considering the factors that

effect the costs and benefits. The modules share common concepts and factors; such as learning rates, task weights, and cost elements; in order to ensure consistent results.

The design of the OSBATS system provides the flexibility required to accommodate the complex interactions involved in training-system design. This approach captures the inherently iterative nature of the training-system design process as described early in this report. The model provides methods that allow the results of analyses using any OSBATS module to provide information used by other modules. Thus, the model's modular structure allows easy repetition and refinement of results.

The characteristic of the OSBATS model that most distinguishes it from its predecessors is its emphasis on training-device design. The training-device design modules of the OSBATS model allow the user to investigate and compare many design options. The design engineer may use the results of these modules to determine which fidelity and instructional feature alternatives should be included in the developing training-device design, based on the training requirements. All other existing training-development models emphasize evaluation. Those models allow the user to evaluate a single training-device design, after it is generated. Application of other models to a large number of alternative designs would be overly burdensome on the design engineer. Thus, the OSBATS model has opened up the most important stage of the training-device design process to the benefits of analytic modeling.

The OSBATS model, unlike most others, aggregates cost and effectiveness estimates to develop recommendations based on a effectiveness/cost ratio. Other models apply a benefit analysis followed by a cost analysis. For example, Kribs, Simpson, and Mark (1983), in their review of media selection models, identified five subtasks that were common to these models. These subtasks included a ranking of training media for training effectiveness followed by a cost tradeoff analysis used to perform the final selection. In a similar fashion, the instructional support feature guidelines developed for the Air Force (Logicon, 1985) specify a benefit analysis followed by an analysis of technology and cost considerations. The integrated effectiveness/cost analysis provided by OSBATS is superior to methods that perform sequential effectiveness and cost analyses, in that the latter methods tend to reject options that offer moderate benefit at a low cost in favor of options that offer high benefit at a high cost. It seems that more overall effectiveness can be obtained within a cost budget with several moderately effective, but inexpensive options, than with one or two highly effective, but expensive options.

In developing the OSBATS model, we have produced the following other advancements in specific areas of training-system modeling.

1. We have extended the framework for training-device optimization initially proposed by Roscoe (1971) to consider the impact of constraints on the use of training device or actual equipment on the time or cost required to meet training requirements.
2. We have developed new procedures to determine task fidelity and instructional feature requirements from descriptions of the activities involved in the tasks.
3. We have developed new methods to cluster tasks according to their needs for simulation and their requirements for a sophisticated simulation capability.

Needs for Future Model Development

The OSBATS model has been completely specified and prototype software has been developed. However, further work is needed before the model is transferred to the user community. The required activities include model expansion, data base development, model calibration and validation, and software enhancement. Some specific needs are outlined below.

Technology Transfer

The ultimate goal of the OSBATS research and development effort is the transfer of the software to the engineers responsible for training-device concept formulation. However, the current version of the system is not sufficiently developed to allow direct transfer to users. Barriers to technology transfer come from both limits in the state of model development, and from the process by which the model was developed.

The current OSBATS data base supports the use of the model over a limited domain. Although the specific domain of application is the AH-1 training course, we think that the model should be applicable with only minor changes to most training domains involving rotary-wing aviation operations. Use of the model outside of this domain requires the user either to collect additional data from subject-matter experts or to make assumptions about the values of such data and suffer a consequent loss in the accuracy of the model's predictions. Furthermore, operation of the model outside of the domain for which it was originally developed will probably require assistance from the model developer, the programmer, or both, to tailor the model to

the new situation. Although our analytical evaluation has indicated that the model processes are general, undoubtedly situations will arise in future applications that require modifications to the model or software.

Nevertheless, it should be possible, with appropriate assistance, to apply the model to a wide variety of problems. We think that the application of the model on an actual training design problem should be a high priority. The model application will establish a working relationship between model developers and model users. The feedback obtained from model users will provide a wealth of information that can be used to improve the model. In addition, we are confident that the model will provide the engineer insights that can be used to produce a better training-device concept.

The initial phase of the OSBATS development process was conducted with limited interactions with the eventual model users. The engineers were used primarily to evaluate the software and to provide information on the procedures currently used for training-device concept formulation. Future development should have a much greater level of involvement by the engineers who will use the OSBATS model in concept formulation. We recommend that future development efforts include a mechanism that will provide an ongoing dialogue between the model developers and potential users to tailor the model to user needs, increase user ownership of the model, and support technology transfer as well as ensure management support.

Other needs for future model development support the need for technology transfer. That is, new model capabilities, more comprehensive data bases, easy data collection and entry procedures, and model calibration and validation will all increase the likelihood of successful technology transfer, as well as offer other enhancements to the quality of the OSBATS model.

Additional Modeling Capabilities

We envision that as the OSBATS model is transferred to the training-device design engineers, many of the requirements for additional modeling capabilities will come from the user. At this stage in the development process, we have received some suggestions from potential users; other ideas have come from our own use of the model. The following list briefly describes several possible enhancements to the OSBATS model's capability.

1. Development of new task clustering methods that reflect other rationale for partitioning tasks, such as similarity of fidelity requirements, mission phase, and so forth. One of the critical early decisions in training-device design specifies the tasks used as the basis of the training-device

design. The current Simulation Configuration Module contains one rationale for clustering tasks. Because of the importance of this decision, we think that a variety of task-clustering methods should be available to the user.

2. Enhancements to model integration capabilities. The OSBATS model currently includes several mechanisms that allow the results of one module to be used in a later module. Additional integration of the modules can improve their usefulness. For example, there is a need to incorporate the simulation requirements determined in the Simulation Configuration Module into the recommendations of the other modules.
3. Expansion of the model to new training technologies. New options that are available to the training-system designer should be evaluated as alternatives to traditional simulation-based methods. Examples of training methods that may require more attention include embedded training, part-task training and skill training. Of course, allowances must also be made for new technology and research information about the use of the technology.
4. Develop ways to incorporate school requirements and constraints into the recommendations of the model. The school may require that certain features be included in a training-device design. Similarly, the school may have constraints on space or time that have an impact on the optimal training-device concept. There is a need for methods that allow all modules to consider these requirements in their analysis.

Data Collection Methods

One of the chief barriers to the application of the OSBATS model to a new training domain is the effort required to obtain the necessary data. There are several activities that could be accomplished to reduce the effort required for data collection. First, standard procedures for data collection should be developed. To the extent possible, these procedures should minimize the requirement for judgments by subject-matter experts. Where precise data are not available, methods for making assumptions about data values should be developed, and the impact of these assumptions on the results of the decision process determined. Finally, procedures should be developed to obtain required data from existing training data bases.

Model Calibration/Validation

The results of the OSBATS model hinge on several key assumptions about learning and transfer processes. Attempts to validate the model should focus on these key assumptions. The validation process will involve both determining the best value for key assumptions (calibration) and testing whether this assumption provides an adequate account of the learning and transfer processes addressed by the model (validation). Because of the large effort required for model validation, we recommend that the validation effort begin with a careful analytical evaluation of the model assumptions to determine which assumptions are the most critical to the model results. The sensitivity analyses conducted under the current effort should provide some guidance in identifying critical assumptions.

Software Enhancements

One of the requirements for technology transfer will be the development of production-quality software representing the OSBATS model. The production version of OSBATS will integrate analytic, rule-based, and data management capabilities of the model. We expect that the next version of OSBATS will incorporate several enhancements to the model software, such as a simplified user interface that is common to all modules, increased access to the logic that is used in rules bases, and access to the data that form the basis of the recommendations of the model. In addition, the next version of the software should incorporate any new analytical and data management capabilities that are developed.

Some software enhancements may be investigated using the current prototype software. Candidate enhancements for development on the prototype system include user interface improvements, more sophisticated help capabilities, and additional or improved displays. Development of these methods on the prototype software allows these methods to be evaluated before they are incorporated into the production software.

References

- Bickley, W. R. (1980). Training device effectiveness: Formulation and evaluation of a methodology. (Research Rep. 1291). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences. (AD A122 777)
- Carter, G., & Trollip, S. (1980). "A constrained maximization extension to incremental transfer effectiveness, or, how to mix your training technologies." Human Factors, 22(2), 141-152.
- Cronholm, J. N. (1985). "The optimization of training systems." Proceedings of Symposium on the Transfer of Training to Military Operational Systems. (pp. 237-259). Brussels, Belgium: NATO.
- Department of Defense Training Data and Analysis Center (1985). Training resources: Macro level training funds prototype data base preliminary results. Orlando, FL: Author.
- Gilligan, E. L., Elder, B. L., & Sticha, P. J. (1988). Optimization of simulation-based training systems: User's guide. (HumRRO Research Product RP-PRD-88-30). Alexandria, VA: Human Resources Research Organization.
- Kribs, H. D., Simpson, A. C., & Mark, L. J. (1983). Automated instructional media selection. (AIMS) (Tech. Rep. NAVTRAEQUIPCEN 79-C-0104-1). Orlando, FL: Naval Training Equipment Center.
- Logicon, Inc. (1985). Instructional support feature guidelines. (AF Contract No. F33615-84-C-0054). San Diego, CA: Author.
- Newell, A., & Rosenbloom, P. (1981). Mechanisms of skill acquisition and the law of practice. In J. R. Anderson (Ed.), Cognitive skills and their acquisition. Hillsdale, NJ: Erlbaum.
- Povenmire, H. K., & Roscoe, S. N. (1973). Incremental transfer effectiveness of a ground-based general aviation trainer. Human Factors, 15, 534-542.
- Roscoe, S. N. (1971). Incremental transfer effectiveness. Human Factors, 15, 534-567.
- Ross, D. T., & Schoman, K. E., Jr. (1977). Structured analysis for requirements definition. IEEE Transactions on Software Engineering. SE-3.
- SofTech, Inc. (1981). Integrated computer-aided manufacturing (ICAM). Architecture Part II. Volume IV - Function modeling manual (IDEF0). Waltham, MA: Author.

- Sprague, R. H., Jr., & Carlson, E. D. (1982). Building effective decision support systems. Englewood Cliffs, NJ: Prentice-Hall.
- Sticha, P. J., Blacksten, H. R., Buede, D. M. (Decision Logistics, Inc.) & Cross, K. D. (Anacapa Sciences, Inc.) (1986). Optimization of simulation-based training systems - Vol III: Model description (HumRRO Final Report FR-PRD-86-13). Alexandria, VA: Human Resources Research Organization.
- Sticha, P. S., Blacksten, H. R., Buede, D. M., Singer, M. J., Gilligan, E. L., Mumaw, R. J., & Morrison, J. E. (1988). Optimization of simulation-based training systems: Model description, implementation, and evaluation (HumRRO Final Report FR-PRD-88-26). Alexandria, VA: Human Resources Research Organization.
- Sticha, P. J., Singer, M. J., Blacksten, H. R., Mumaw, R. J., Buede, D. M. (1987). Optimization of simulation-based training systems - Year II report (HumRRO Interim Report IR-PRD-87-30). Alexandria, VA: Human Resources Research Organization.
- Sticha, P. J., Singer, M. J., Blacksten, H. R., Morrison, J. E., Cross, K. D. (1988). Research and methods for simulation design: State of the Art (HumRRO Final Report FR-PRD-88-27). Alexandria, VA: Human Resources Research Organization.
- Stock, J. A., Gilligan, E. L., Sticha, P. J. (1988). Optimization of simulation-based training systems: Programmer's Guide (HumRRO Research Product RP-PRD-88-29). Alexandria, VA: Human Resources Research Organization.
- Willis, R. P., (1988). The Optimization of Simulation-Based Training: Model Data Collection and Utilization. (Final Report, Contract N61339-86-0009-0002), Eagle Technology, Inc., Winter Park, FL.