

PAYLOAD TRAINING METHODOLOGY STUDY (PTMS)

FINAL REPORT

Space Station Payload Training Program Design Contract

Contract Number NAS8-3773

February 1990

Prepared For:

**MAN SYSTEMS INTEGRATION BRANCH (EL15)
Operations Development Division
Systems Analysis and Integration Laboratory
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812**

Prepared By:

**ESSEX CORPORATION
Space Systems Group
690 Discovery Drive
Huntsville, Alabama 35806**

TABLE OF CONTENTS

List of Tables	iv
List of Figures	v
List of Acronyms	vi
Abstract	vii
INTRODUCTION	I-1
1.0 PROGRAM CONCEPT	1-1
1.1 Purpose	1-1
1.2 Scope	1-1
1.3 Training Needs Assessment	1-1
1.3.1 Experiment Database Development	1-2
1.3.2 Task Hierarchy Development	1-2
1.3.3 Training Objective and Test Development	1-3
1.4 Instructional Plan Development	1-3
1.4.1 Instructional Methods and Strategies	1-4
1.4.2 Instructional Media Selection	1-4
1.4.3 Hands-on Media Functional Requirements and Functional Specifications	1-5
1.4.4 Syllabi Development	1-6
1.5 Simulator Requirements Derivation	1-8
1.5.1 Experiment Overview Report (EOR)	1-8
1.5.2 Simulator Approach Synthesis	1-9
1.5.3 Simulator Top-Level Requirements Document	1-10
1.5.4 Experiment Simulator Requirements Document	1-10
1.6 Payload Training Systems Verification	1-11
1.6.1 Increment-Independent Verification Planning	1-11
1.6.2 Specification Verification	1-12
1.6.3 Verification Testing	1-12
1.7 Validation	1-13
1.7.1 Academic Instructional Validation	1-14
1.7.2 Hands-On Media Validation (Including Simulators)	1-16
1.7.3 Ongoing Validation	1-16
1.8 Training Program Template	1-16
2.0 TRAINING NEEDS ASSESSMENT METHODOLOGY (#1)	2-1
2.1 Analysis of System Requirements	2-1
2.1.1 Experiment Database	2-1
2.2 Analysis of Training Requirements	2-6
2.2.1 Construction of a Task Hierarchy	2-7
2.2.2 Assignment of Task Attributes	2-12
2.2.3 Task Classification	2-14
2.2.4 Training Objective and Test Development	2-17
2.3 Methodology Summary	2-27

3.0	INSTRUCTIONAL PLAN DEVELOPMENT METHODOLOGY (#2)	3-1
3.1	Instructional Methods and Media (Defining the Active Learning Environment)	3-1
3.1.1	Instructional Methods and Strategies	3-3
3.1.2	Instructional Media Selection	3-8
3.1.3	Instructional Methods and Media Procedures Summary	3-19
3.2	Hands-On Media Functional Requirements and Functional Specifications	3-19
3.2.1	Hands-On Media Functional Requirements	3-23
3.2.2	Hands-On Media Functional Specifications	3-46
3.2.3	Simulator Functional Requirements and Functional Specifications Procedures Summary	3-47
3.3	Syllabi Development	3-48
3.3.1	Lesson Organization	3-50
3.3.2	Instructional Requirements	3-54
3.3.3	Lesson Design	3-56
3.3.4	Lesson Specifications	3-58
3.3.5	Instructional Materials Development	3-59
3.3.6	Syllabi Development Procedures Summary	3-60
4.0	SIMULATOR REQUIREMENTS DERIVATION METHODOLOGY (#3)	4-1
4.1	Experiment Overview Report (EOR)	4-4
4.1.1	Purpose	4-4
4.1.2	Structure	4-5
4.1.3	Experiment Overview Report Procedures Summary	4-8
4.2	Simulator Approach Synthesis	4-8
4.2.1	Hands-On Media Functional Specification Review	4-11
4.2.2	PTC to SSTF Coordination	4-16
4.2.3	Preliminary Simulator Approach	4-16
4.2.4	Increment Training Plan	4-22
4.2.5	Simulator Approach Synthesis Procedures Summary	4-28
4.3	Simulator Top-Level Requirements Document	4-28
4.3.1	Document Assembly	4-30
4.3.2	PI Interview	4-33
4.4	Experiment Simulator Requirements Document	4-35
4.5	Simulator Top-Level and Detailed Requirements Derivation Procedures Summary	4-37
5.0	TRAINING DEVELOPMENT VERIFICATION METHODOLOGY (#4)	5-1
5.1	Introduction	5-1
5.1.1	Verification Definition	5-1
5.1.2	Levels of Verification	5-2

	5.1.3	Verification Options	5-3
	5.1.4	Depth of Verification	5-4
5.2		Specification Verification	5-5
	5.2.1	Training Objectives Verification	5-6
	5.2.2	Instructional Plan Verification	5-8
	5.2.3	Simulator Requirements Verification	5-10
	5.2.4	Design Verification	5-14
	5.2.5	Implementation Verification	5-15
5.3		Verification Testing	5-16
	5.3.1	Purpose	5-16
	5.3.2	Responsibilities and Deliverables	5-16
	5.3.3	Methods for Test Documentation Production	5-17
	5.3.4	Test Execution Activities	5-18
5.4		Summary	5-19
6.0		TRAINING SYSTEM VALIDATION METHODOLOGY (#5)	6-1
	6.1	Academic Instructional Validation	6-4
		6.1.1 Academic Instruction Test Plans	6-4
		6.1.2 Conduct of Academic Validation	6-6
	6.2	Hands-On Media Validation (Including Simulators)	6-7
		6.2.1 Hands-On Media Test Plans	6-8
		6.2.2 Conduct of Hands-On Media Validation	6-10
	6.3	Ongoing Validation	6-11
		6.3.1 Performance Measures for Ongoing Validation	6-11
		6.3.2 Conduct of Ongoing Validation	6-12
	6.4	Engineering Change Requests	6-14
7.0		SIMULATOR FIDELITY DEFINITIONS	7-1
	7.1	Terminology	7-1
	7.2	Functionality	7-3
	7.3	Hardware Classes	7-3
	7.4	Planned Experiment Simulator Usage at the PTC	7-4
8.0		REFERENCES	R-1
		APPENDIX A	A-1
		APPENDIX B	B-1
		APPENDIX C	C-1
		APPENDIX D	D-1
		APPENDIX E	E-1
		APPENDIX F	F-1

LIST OF TABLES

Table 3-1.	Classification of General Learning Types Applicable to Payload Training	3-4
Table 3-2.	Definition and Classification of Instructional Methods Applicable to Payload Training	3-5
Table 3-3.	Hands-On Versus Academic Media Classification	3-10
Table 3-4.	Applicability of Instructional Media to Learning Types	3-12
Table 3-5.	Representative Range of Instructional Media Suitable for Payload Training	3-15
Table 3-6.	Variables that Interact with Fidelity	3-22
Table 3-7.	Eleven Types of Elemental Learning Tasks	3-37
Table 3-8.	CFD Rating Potential for Each Information-Processing Stage by Learning Task	3-42
Table 7-1.	Simulator/Trainer/Mockup Classification Matrix	7-2

LIST OF FIGURES

Figure 1-1.	Training Program Development Template	1-15
Figure 2-1.	Experiment Database	2-3
Figure 2-2 (a).	Sample Task Hierarchy	2-10
Figure 2-2 (b).	Sample Task Listing	2-11
Figure 2-3.	Sample Task Sheet	2-16
Figure 2-4 (a).	Objective Worksheet	2-20
Figure 2-4 (b).	Objective Worksheet	2-21
Figure 2-4 (c).	Objective Worksheet	2-22
Figure 2-4 (d).	Objective Worksheet	2-23
Figure 2-5.	Breakdown from a Terminal Objective	2-24
Figure 3-1.	Instructional Plan Development	3-2
Figure 3-2.	Selection of Instructional Media	3-7
Figure 3-3.	Instructional Media Selection	3-18
Figure 3-4.	Functional Specifications and Requirements Derivation and Allocation to Media Types	3-20
Figure 3-5.	Sample Format for Task Analysis Data Used in Fidelity Determinations	3-25
Figure 3-6.	Sample Format for Simulator Functional Requirement Form	3-30
Figure 3-7.	Sample Matrix for a Fidelity Study of Task Difficulty at Various Training Levels	3-33
Figure 3-8.	Accomplishment of a Task/Objective in Terms of Its Information-Processing Demands	3-35
Figure 3-9.	Syllabi Development	3-49
Figure 4-1.	Inputs/Outputs of Simulator Requirements Derivation Process	4-3
Figure 4-2.	Development Flow for Simulator Approach Synthesis	4-9
Figure 4-3.	Hands-On Media Functional Specification Review Process	4-15
Figure 4-4.	"Typical" Payload Experiment Configuration	4-18
Figure 4-5.	PTC Increment Planning from Training Development Input	4-23
Figure 4-6.	Relations Between All Levels of Objectives, Training Media, and PTC Trainers	4-25
Figure 4-7.	Assembling the Simulator Top-Level Requirements Document (STLRD)	4-29
Figure 5-1.	Training Needs Assessment	5-7
Figure 5-2.	Instructional Plan Development Traceability	5-9
Figure 5-3.	Simulator Requirements Verification Traceability	5-12
Figure 6-1.	Simplified Training Program Schedule for Validation	6-3

LIST OF ACRONYMS

AIMS	Automated Instructional Media Selection
ATP	Acceptance Test Procedure
CAA	Computer-Aided Analysis
CBT	Computer-Based Training
CDR	Critical Design Review
CFD	Task Criticality, Frequency, Difficulty
CVA	Concurrent Verification Activities
DEP	Dedicated Experiment Processor
ECR	Engineering Change Request
EOR	Experiment Overview Report
ESRD	Experiment Simulator Requirements Document
ICD	Interface Control Document
ISD	Instructional Systems Development
JPR	Job Performance Requirement
JSC	Johnson Space Center
MSFC	Marshall Space Flight Center
PCTC	Payload Crew Training Complex
PDR	Preliminary Design Review
PED	Payload Experiment Development
PI	Principal Investigator
POIC	Payload Operations Integration Center
PRR	Preliminary Requirements Review
PTC	Payload Training Center
PTT	Part-Task Trainers
SAD	Simulation Approach Document or Simulator Approach Document
SAR	Simulator Acceptance Review
S-R	Stimulus-Response
SSE	Software Support Environment
SS	Space Station
SSP	Space Station Program
SSTF	Space Station Training Facility
STAR	Simulator Training Acceptance Review
STLRD	Software Top Level Requirements Document
TR	Training Requirement
TRDS	Training Requirements Development System

CR-183907
Dane Scott EL15

Space Station Payload Training Program Design Contract
Contract Number NAS8-37737

Payload Training Methodology Study

Final Report
February 1990

Prepared For:

MAN SYSTEMS INTEGRATION BRANCH (EL15)
Operations Development Division
Systems Analysis and Integration Laboratory
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

Prepared By:

ESSEX CORPORATION
Space Systems Group
690 Discovery Drive
Huntsville, Alabama 35806

ABSTRACT

This final report documents the results of the Payload Training Methodology Study (PTMS). This report defines methods and procedures for the development of payload training programs to be conducted at the Marshall Space Flight Center Payload Training Complex (PCT) for the Space Station Freedom program.

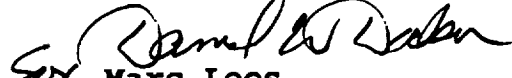
The study outlines the overall training program concept as well as the six methodologies associated with the program implementation. The program concept outlines the entire payload training program from initial identification of training requirements to the development of detailed design specifications for simulators and instructional material.

The following six methodologies are covered in this final report:

1. The Training and Simulation Needs Assessment Methodology defines the methodology of the initial assessment of training needs to support individual experiment, integrated experiment, and integrated simulation training.
2. The Simulation Approach Methodology defines the process for establishing a simulator design approach.
3. The Simulation Definition Analysis Methodology describes a Systems Engineering process of requirements derivation which will define proper and complete functionality for training.
4. The Simulator Requirements Standardization Methodology defines a standard to establish, define, develop, test, review, analyze, update, and finalize simulator requirements.
5. The Simulator Development Verification Methodology is a method to perform verification of the requirements and products derived during the simulator development.
6. The Simulator Validation Methodology discusses the validation of the developed simulator to show that simulator requirements to support training have been fulfilled.

This effort has been performed by the Essex Corporation Space Systems Group and TRW Systems Development Division under MSFC contract NAS8-37737.

Prepared By:


Marc Loos
PTMS Lead Engineer

Approved By:


Daniel W. Dodson
Project Manager

INTRODUCTION

Study Purpose

The purpose of this study is to develop methods and procedures for the development of payload training programs. These methods and procedures are directed towards the needs and concerns of training to be conducted at the Payload Training Complex (PTC) at Marshall Space Flight Center for the Space Station Freedom program. NASA plans not only to develop methods and procedures that can be used to construct a training development system but also to explore ways in which this system could be automated and improved over the life of the Space Station Freedom.

Study Outputs

The methods and procedures developed here are collectively referred to as the Training Requirements Development System (TRDS). This system has been organized into five methodologies, roughly corresponding to Tasks 2-7 of the Statement of Work (SOW) for the PTMS. In addition, there is a Program Concept document, which corresponds to Task 1, and a treatment of simulator fidelity definitions corresponding to Task 8.

The issue of TRDS automation and upgrading has been addressed in a Trade Survey. The results of this survey are included in this report in presentation form.

Discussion of Conclusions

Task 1 - Program Concept

This task is to generalize and outline the TRDS from the initial identification of training requirements to the development of detailed design specifications for simulators and instructional materials. This requirement is interpreted as an overview of the TRDS. It outlines its methods and procedures in a conceptual manner. A program concept has been written which satisfies this requirement. It comprises Section 1.0 of the PTMS Final Report.

Task 2 - Training and Simulation Needs Assessment Methodology

Task 2 requires a methodology for initial assessment of training needs to support an individual experiment, integrated experiment, and integrated simulation training. A methodology was developed which organizes experiment data into a convenient format for analysis, analyzes the experiment data to derive tasks-to-be-trained, and develops training objectives (for the tasks) which identify all skills and knowledge to be trained.

The methodology specifically addresses the development of training objectives which address integrated payload and integrated simulation training needs as well as the needs for individual experiment training. Since this methodology makes no distinction between training needs and simulation needs, it has been renamed the "Training Needs Assessment Methodology." It comprises Section 2.0 of the PTMS Final Report.

Task 3 - Simulation Approach Definition Methodology

Task 3 requires a methodology which would develop a simulator definition, based on training needs, objectives, resources availability, etc. A process for establishing an integrated payload, as well as individual experiment requirements.

These requirements were perceived as too narrowly-focused on simulators, rather than on the entire training process which involves both academic and hands-on instruction. Therefore, with Program permission, a methodology was developed which expands on those requirements to include the development of instructional materials to support both hands-on and academic training. Using the previously developed Training Objectives and Tasks as primary inputs, this methodology derives training methods and media for all objectives, and develops hands-on media Functional Specifications and Lesson Specifications. The Functional Specifications address the initial requirement for a simulator design approach, while the Lesson Specifications do the same for the academic materials. A more detailed treatment of a simulator design approach has been allocated to the next methodology.

Because this methodology's scope has been expanded beyond what was originally requested, it has been renamed the "Instructional Plan Development Methodology." It comprises Section 3.0 of the PTMS Final Report.

Task 4 - Simulation Definition Analysis

Task 4 requires a method to determine simulator requirements which will define proper and complete functionality for training. A methodology was developed for this, which describes a Systems Engineering process of requirements derivation and management. Hands-on media Functional Specifications developed during Instructional Plan Development are refined by considerations of integrated experiment requirements, integrated simulation requirements and PI-provided training objectives. A simulator approach is developed, based on situational factors and experiment design features. A top level simulator requirements document is assembled which maps simulator functional requirements onto the structure defined by the simulator

approach. Finally, a detailed requirements document is constructed through further detailing of the top-level requirements. To clarify the purpose for this methodology, it has been renamed the "Simulator Requirements Derivation Methodology" and comprises Section 4.0 of the PTMS Final Report.

Task 5 - Simulator Requirements Standardization

Task 5 requires standardized approach methods and procedures to establish, define, develop, test, review, analyze, update, and finalize simulator requirements. As such, this task is seen as a requirement for standardized procedures throughout the training development process, rather than a discrete methodology in and of itself. Therefore, this requirement has been satisfied by the following principles, used throughout the TRDS and discussed herein:

- a) Changes to training, or to training products in development, will be addressed (as discussed in the Validation Methodology) initially with a change assessment. This assessment is made to determine what must be updated on the training product (simulator, workbook, script, etc.) as well as to simulator requirements documentation. Once the necessary change has been defined and approved, a two-pronged approach will be used to deal with it. First, the change will be implemented on the training product immediately to minimize impact to training or training development. Secondly, an activity is initiated to update as appropriate, all development documentation affected by the change. The change request, documented by an Engineering Change Request (ECR) form, will remain open until all change activity has been approved.
- b) For Verification, Validation, and Configuration Management purposes, traceability will be established by direct references between distinct elements of the TRDS process. An Experiment Database will contain clearly defined data items to which all development products can be traced. It should be possible to draw lines from specific experiment information to discrete elements of the Task Hierarchy, Functional Specifications, Lesson Specifications, and simulator requirements documents.
- c) Outlines and examples of documents, forms, and other data items related to the development process have been given to help illustrate the concept of standardization in the process. Due to the uncertainties which abound in this stage of the Space Station Freedom Program, it would be fruitless to prescribe exact details for the implementation

of these methodologies. The level of detail given should be sufficient to explain TRDS methods and procedures, and provide a starting point for their practical implementation.

- d) The Automated Tools Trade Survey discusses the ways in which an automated system based on a relational database can contribute to procedures standardization. Specifically, automation of the TRDS process will encourage and enforce consistency in the development products and in the procedures used to develop those products.
- e) The Validation and Verification Methodologies describe procedures and methods for testing, reviewing, analyzing, updating, and finalizing simulator requirements.
- f) The Training Assessment, Instructional Planning, and Simulator Requirements Derivation methodologies present methods and procedures for establishing, defining, and analyzing training requirements.

Task 6 - Simulator Development Verification Methodology

This requirement is for a methodology to perform verification on the products derived during simulator development. This scope has been expanded to encompass related items (such as academic materials and training scripts). Verification in this instance is defined to include all processes performed in order to prove that PTC-hosted training requirements are being properly implemented during training development. This methodology comprises Section 5.0 of the PTMS Report.

Task 7 - Simulator Validation Methodology

This requirement is for a methodology to perform validation on the developed simulator to show that simulator requirements have been fulfilled. As with the Verification Methodology, this scope has been expanded to include validation of all devices and materials to be used during training. This includes simulators, scripts that support simulator training, academic lessons, and instructional materials (such as workbooks, exhibits, flipcharts, etc.). In this instance, validation is defined to include all processes performed for each Space Station Freedom experiment so that it can fulfill its overall training objectives. This methodology comprises Section 6.0 of the PTMS Report.

Task 8 - Simulator Fidelity Definitions

This requirement is to establish a method for classifying simulators according to their level of fidelity. It was further required that these classifications be used to describe the simulators' functional physical interface necessary to achieve training and simulation objectives.

A method of classifying simulators was developed jointly with JSC in the course of the PTC Simulation Computer System (SCS) Study. Since it was expressly developed in order to define a common nomenclature for the Space Station Freedom development community, this method is adopted here. The classification system was then used to define the required levels of simulator fidelity and functionality necessary to achieve the most likely configurations for payload training at the PTC. These definitions will be included as Section 7.0 of the PTMS Final Report.

Training Metrics

Essex was also asked to explore the implication of metrics for payload training performance evaluation. Essex responded with a study of performance measures, their derivation, validation and use. This study is provided in Appendix B of the PTMS Report.

Automated Tools Survey Report

In addition to the Tasks listed in the SOW, Essex was asked to explore possibilities for automation of the TRDS. In response, Essex performed a survey of training analysis automation tools and Computer Aided Software Engineering (CASE) tools. The results of the study are provided in Appendix D of the PTMS Report. They consist of 1) a presentation of available tools, and 2) a directory-style listing of the tools complete with the companies producing them.

PTMS Issues

During the early stages of the PTMS, various training development issues were discussed and debated. The PTMS handled these issues by producing mini-reports on them. These mini-reports are included for historical reference in Appendix E of the PTMS Report.

TRDS Briefing Charts

A set of briefing charts, which summarizes the key elements of the Training Requirements Development System are included. These charts are provided as Appendix F of the PTMS Report.

Conclusion

This study has generated methodologies and references to enable the establishment of a systematic, step-by-step program for the development, verification and validation of complete training systems. Furthermore, it has been found that automation of such a system is both feasible and practical. This conclusion is based on a comparison of TRDS processes with commercially available automation tools.

1.0 PROGRAM CONCEPT

The Program Concept is a conceptual outline of the payload Training Requirements Development System (TRDS) from the assessment of top level training needs to the specification of detailed training requirements. These activities are presented in a series of methodologies, discussed here in an overview-type format.

1.1 Purpose

The purpose of the TRDS is to identify training objectives and establish detailed requirements for the design and development of experiment simulators, courseware, instructional materials, and syllabi sufficient to keep pace with the Payload Training Complex (PTC) training cycle (PTC requirements are in turn, driven by the Space Station Program [SSP] launch schedule, proposed payload manifests and other programmatic parameters). As part of training development, the TRDS will perform Verification of training systems in development and Validation of the finished product. Wherever advantageous, it will incorporate software utilities to mechanize and streamline the requirements development process.

1.2 Scope

The TRDS will apply systematic approaches to all aspects of payload training requirements development, ranging from training needs/objectives identification to training implementation. TRDS responsibilities begin with the gathering of early experiment data from the Principal Investigator (PI) and continue throughout the lifetime of the experiment. They include the development of training aids, courseware, and training strategies as well as experiment simulators.

The TRDS will develop requirements for all modes of Marshall Space Flight Center (MSFC) payload related training, ranging from one person - one experiment training to multi-person - one experiment training, to training for entire mission scenarios involving flight crew and ground crew. Systems training will be covered only so far as is necessary to facilitate training scenarios where systems interaction occurs. Likewise, Payload Operations Integration Center (POIC) training will be considered only so far as it is necessary to facilitate the training of payload operations.

1.3 Training Needs Assessment

Training Needs Assessment is the first phase of the TRDS. The purpose of this phase is to derive training requirements for both academic and hands-on media training. These training

requirements will be derived from specific information about the experiment to be trained, and the policies and constraints imposed by the Marshall Payload Training Program. In many ways, this is the most critical phase, since all subsequent development steps will draw upon the results reached here.

The Training Needs Assessment process uses disciplined Instructional System Development (ISD) procedures to perform what is essentially the training program front-end analysis. Training requirements are derived in the form of the tasks necessary to operate and maintain a particular experiment. These tasks are characterized and classified for training in a way which will provide source data for all other development steps in the TRDS. Training Objectives and test criteria for the accomplishment of each objective are then developed from the tasks to be trained.

1.3.1 Experiment Database Development

The first stage of Training Needs Assessment involves obtaining sufficient information about the experiment to be trained to allow top-level simulation and training system requirements to be developed. The training analyst will develop an in-depth understanding of experiment functions and interfaces by gathering experiment requirements into an Experiment Database. These experiment requirements will be organized into data items, formatted to be directly usable by specific TRDS processes. They will be maintained as the source for traceability from experiment information, through intermediate products, to detailed academic and hands-on media requirements.

The data items will include:

- (a) Experiment Description
- (b) Experiment Purpose
- (c) Drawings, Schematics and Associated Lists
- (d) Experiment Training Requirements
- (e) Experiment Operational Requirements
- (f) Experiment Operational Requirements
- (g) Experiment Development Schedule
- (h) PI Training Plan
- (i) SSP Training Plan
- (j) Experiment Review Materials
- (k) NASA and PTC Training Policies
- (l) Simulator Development Schedule
- (m) Trainee Information

1.3.2 Task Hierarchy Development

Once a body of knowledge has been assembled about the operation of an experiment, this knowledge may be analyzed to derive the tasks necessary to operate and support the experiment during a

Space Station increment. These Tasks may be organized into a Task Hierarchy consisting of Activities, Phases, Tasks, and Sub-Tasks. This hierarchical arrangement demonstrates proper task sequencing and the dependent relationships between tasks and levels of tasks.

After a Task Hierarchy has been established, the tasks are characterized and classified in various ways. Task attributes such as Conditions and Standards of Performance are added to them, as well as a number of other properties such as Criticality and Difficulty. When each task has been sufficiently detailed, an Objective Hierarchy is derived from the Task Hierarchy. The Objective Hierarchy represents the behaviors which are to be trained in order to accomplish the tasks. Finally, Criterion Tests and Diagnostic Tests are derived for each Training Objective.

1.3.3 Training Objective and Test Development

After the Task Hierarchies for an experiment are defined, they can be used to develop Objective Hierarchies. Each Objective Hierarchy is comprised of Training Objectives, Criterion Objective Test, and Diagnostic Test. In contrast to the Task Hierarchy which states what must be done to operate the experiment, the Objective Hierarchy describes what must be learned in order to perform experiment tasks. The training objectives will be used as the framework for all instruction, both academic and hands-on. Lessons will be designed around accomplishment of the objectives and simulators will be designed with the functionality and fidelity necessary to train the specified objective behaviors. Objectives will be used to determine lesson sequence and aid in training media selection.

Criterion Tests and Diagnostic Tests will be derived for each objective to evaluate students' accomplishment of the objectives and will be used for final Validation of the total training system. As a check, developed objectives will be compared against objectives previously identified by the PI, to spot omissions and contradictions (if any). Once the training Objectives have been specified, media allocations may be made based upon them and upon the previously developed task attributes. The result of Training Objective and Test Development is a hierarchy of objectives with related Test Items. These Objectives and Tests identify what is to be taught, and how the results of this teaching will be demonstrated.

1.4 Instructional Plan Development

Instructional Plan Development refers to a set of processes which define how developed instructional requirements are met, as well as how training effectiveness is to be measured. In so doing,

these processes produce media functional specifications, lesson specifications, test plans, and sequenced lesson plans. The major inputs to Instructional Planning are the comprehensive hierarchies of behavioral objectives and related test items produced during Training Objective Development. These hierarchies and tests specify the Terminal Objectives and Component Enabling Objectives, skills, and knowledges for every task to be trained. During Instructional Plan Development, these objectives are allocated to training media, analyzed for their functional requirements, organized into lessons, and sequenced according to an overall instructional strategy. Inputs which aid these processes include PTC training guidelines and policies, resource constraints, crew position requirements, and trainee individual and group characteristics. Outputs from this effort allow both simulator (hands-on) and academic media to be developed, as well as the supporting instructional aids and materials.

Objectives and tests identify what is to be taught. Instructional Planning determines how it will be taught, and how to determine if the instruction is effective.

1.4.1 Instructional Methods and Strategies

Once the training objectives have been defined for an experiment, along with the underlying skills and knowledges required, instructional planning can begin by choosing the methods to be used in teaching them. The most straight-forward way to determine optimal instructional methods for a given objective is to relate the behaviors involved to one or more types of learning. Since some instructional methods (and media) are more effective than others in aiding each type of learning, the types of learning involved in reaching an objective can help determine appropriate instructional methods for that objective. While there is no specific formula relating learning types to optimal instructional method, a range of suitable candidate methods can be intuitively determined in this manner. From the initial group of candidate instructional methods, a further selection may be made by consideration of factors such as student individual and group characteristics, cost, and resources.

The output of the methods selection process will be a set of learning types and candidate instructional methods stored as attributes of each objective, in the Experiment Database.

1.4.2 Instructional Media Selection

Besides instructional strategies, the most important aspect of the active learning environment is the medium, or means through which the student will be given information. These means can range from classroom lecture, to a workbook, to simulators or

training on actual system (payload) equipment. Appropriate media must be selected for each objective based primarily on training effectiveness for that objective. Evaluation of the effectiveness of a medium must consider the ways in which it will accommodate presentation of the information, use or practice, and feedback to the student. If alternative media have equal effectiveness in each of the preceding areas, then the choice between them should be made on the basis of cost, availability, maintainability, or other external factors.

Candidate media are established for each objective by relating characteristics of the instructional requirements which they represent to attributes of the media alternatives. This relational process, however performed, is a prime candidate for proceduralization and automation. A number of automated models have been proposed and developed, such as the Automated Instructional Media Selection (AIMS) system developed by Kribs, Simpson, and Mark (1983). The AIMS system is designed to relate up to 90 instructional characteristics, such as strategy, crew interaction, or degree of feedback, to up to 90 instructional media. The methodology presented in this study is a manual one which is given primarily for illustrative purposes. The most important factor in media selection however, whatever the methodology, is that it be based on instructional requirements and training needs.

1.4.3 Hands-On Media Functional Requirements and Functional Specifications

As a preliminary step towards establishing media functional specifications, each objective will be analyzed separately to determine the functional requirements that will be used later to establish the functional specifications for each media type employed in training. Inputs to functional requirements include Task Analysis data (previously developed), as well as Lesson Specifications which will be generated as part of Syllabi Development.

A third input which is very important to the development of training device requirements is empirical data on the ways in which factors both extrinsic and intrinsic to the training task interact with the device characteristics needed for cost-effective training. These factors include task difficulty, trainee sophistication, task type, etc. Empirical data on these relationships as they specifically relate to payload training are scarce. While the functional requirements derivation process described in Instructional Plan Development (Section 3.0) should provide a reasonable first cut at how to effectively train for specific tasks, systematic efforts to relate training effectiveness to specific instructional strategies and device

features will be necessary if the methodology is to evolve and achieve optimal results in the payload training application.

Once the functional requirements for each hands-on training objective and candidate training medium have been defined, they are examined collectively to establish simulator categories based on similar requirements. These trainer categories are established on the basis of the media candidates for each objective, stage of training, overall instructional strategy, and level of fidelity required. The output of this step shall be collective functional characteristics which will serve to define various levels of hands-on media fidelity or functionality. Functional Specifications are then developed for each of the required hands-on media.

1.4.4 Syllabi Development

With the establishment of candidate methods and media for each training objective, and the development of media functional specifications, the active learning environment should be well defined. At this point then, the basic learning structure may be detailed as to the content and organization of the curriculum. Objectives are clustered into lessons, and sequenced within each lesson to optimize skill and knowledge acquisition. Lesson specifications are written, documenting instructional breadth, depth, methods, and media for subsequent development. Separate training tracks are established for each crew position (for example, Mission Specialist), from sequences of lessons. Figure 3-9 illustrates the Syllabi Development process.

Lesson Organization and Sequencing: Lessons are outlined for each subject matter topic, covering one or more training objectives. The coverage of each lesson should be managed in order to encompass enough material to result in a significant learning yet be restricted to a single topic. Each lesson should include a test which will demonstrate that the student understood the material. Where possible, the lessons should be modularized in such a way as to allow flexibility in course pacing for individuals. Lesson sequencing is performed in a way which shows relationships between activities, avoids duplication, or gaps in training, and promotes an orderly building of skills and knowledges.

Lesson Specification: The Lesson Specification consists of a detailed outline containing or referencing all information necessary to allow writing the actual lesson and developing instructional materials. Lessons will be developed for both academic and hands-on media.

Academic Lesson Specifications: Each specification contains both general lesson information and specific information on each

objective covered in the lesson. General information includes a hierarchical "map" of the lesson objectives, a lesson introduction, overall instructional strategy, student prerequisites, and a description of the instructional materials required to conduct the lesson. Specific information on each objective includes the objectives themselves, along with their associated Conditions and Standards of Performance.

Hands-On Lesson Specifications: These are specifications developed for each lesson to be conducted on a trainer or the actual equipment. Each specification contains the elements required for student practice and instructor evaluation of the objectives in the lesson. These consist of the same items as detailed for academic lessons as well as an outline of tasks to be performed, a description of the instructor guidance to be provided, and references to the academic lessons which support accomplishment of the current objectives.

Evaluation Measures and Mechanisms: Each lesson specification will also include general and specific evaluation procedures. These include tests for each objective, as well as Performance Measures for the entire lesson and curriculum. The objective test items measure the specific behavior associated with that objective, and are derived directly from the test developed during the formulation of the Training Objectives. The Performance Measures are more concerned with overall training effectiveness and lesson and curriculum goals. Their derivation must begin with a clear understanding of the various purposes for evaluation and end with a validation of the derived measures against accepted metrics' criteria for each evaluative purpose.

Instructional Materials Development: The Instructional Materials Development activity receives as input, the functional specifications for all academic media, including Computer-Based Training (CBT) courseware, and lesson specifications for both academic and hands-on media. Its output consists of CBT courseware, workbooks, tests, charts, study guides, training scripts, films, slides, and all other materials necessary to support academic and hands-on training. Academic media materials will be developed first, while hands-on media materials development will wait until after simulator requirements are delineated at Preliminary Design Review (PDR).

At Simulator PDR, the academic instructional materials will be verified for traceability to Instructional Requirements specified in the Instructional Plan. After PDR, with simulator functionality specified, development can proceed for those materials which will directly support the use of experiment simulators for training. Resultant course materials will be presented and reviewed at CDR, in conjunction with designs for the simulators they are intended to support. After CDR,

instruction will begin 15 months before launch using the classroom and CBT materials. Experiment simulator materials will see their initial use (and final testing) during Acceptance Verification and Validation when the simulators are used in the execution of training scenarios.

1.5 Simulator Requirements Derivation

Simulator Requirements Derivation is the process whereby detailed simulator hardware and software requirements are produced which reflect Mission and Science, as well as individual and integrated experiment training objectives. Its primary inputs consist of PI-provided experiment data and hands-on media Functional Specifications. The process, however, must also take into account overall SS training plans, PTC resources, experiment development schedules, and the planned training curricula for each experiment.

The Training Analysis methodologies (#1 & #2) fulfill the role of Instructional Systems Development (ISD) in producing requirements for complete training systems. Simulator Requirements Derivation (Methodology #3) is a Systems Engineering process designed to use these training requirements to formulate simulator requirements. These requirements will in turn, be used as the basis for simulator design and development.

The Simulator Requirements Derivation process can be defined in terms of the data items which will be generated by the developer while deriving simulator requirements. These data items include a) an Experiment Overview Report (EOR), b) a Simulation Approach Document (SAD) for each experiment simulator, c) a description of training scope for each experiment, to coordinate with JSC, d) a Software Top Level Requirements Document (STLRD) for each simulator, and e) a detailed math model and requirements document for each simulator (Experiment Software Requirements Document [ESRD]). Simulator Requirements Derivation, though discussed as a sequential process is actually iterative in nature; gradually producing mature simulator requirements as understanding of particular experiments grows and experiment data becomes available.

1.5.1 Experiment Overview Report (EOR)

The EOR represents an initial effort to evaluate an experiment in terms of the simulation and training problems which it represents. Its building blocks are comprised of data items developed as part of the data acquisition phase of the Training Needs Assessment Methodology (#1). These data items have been designed to fulfill the needs of both the Training Analysis and simulator Systems Engineering processes. Therefore, under ideal circumstances, most of the work involved in producing an EOR will

already have been done for training analysis, and stored in the Experiment Database. If not, the data items must then be derived from experiment information and stored in the Experiment Database, as described in the procedure for Training Needs Assessment (Section 2.1.1). In addition, if the experiment data has changed or been augmented since the time that the data items were developed, it may be necessary to update them before proceeding with further analysis. Any further data items developed as part of Simulator Requirements Derivation should also be included as part of the database so that all analysis efforts will have access to the same inputs.

1.5.2 Simulator Approach Synthesis

Simulator Approach Synthesis is a process which examines the training requirements derived from front-end training analysis for each experiment, and integrates them with each other and with real-world constraints such as PTC policies, status of experiment development, cost-effectiveness strategies, and other external factors. The output of this integration, or synthesis is a preliminary approach for each simulator, documented in a Simulator Approach Document (SAD) for each simulator that will be used to train an experiment in a mission increment. This approach will be an input for the development of top-level simulator requirements and will serve as a generalized game plan for all requirements definition and related activities. As a side-product, the synthesis process will produce a revised hands-on media Functional Specification for each simulator. In so doing, it will also unify all the training objectives for an experiment simulator into an integrated conceptual whole, which can be communicated to JSC for inter-center training coordination.

The products of this process (and earlier ones) will also be useful in coordinating simulator development efforts between the PTC and the PIs. The EOR will flag significant training scope and design details of PI-developed simulators to PTC developers. The PI in turn will receive guidance to ensure that:

- (a) The simulators will be supportable by standard PTC facilities.
- (b) The simulator will satisfy integrated simulator requirements.
- (c) The simulator's coverage of experiment training objectives will complement coverage supplied by the PTC.

This guidance will ideally be embodied in the form of the hands-on media Functional Specification for each simulator; listing all the simulator functional requirements necessary to satisfy

the training objectives allocated to it. PTC interface requirements will be specified by an ICD (to be supplied by PTC programmatic sources). If the finalized Functional Specification is not available early enough to aid PI simulator development, its component parts can be supplied instead. These would consist of preliminary Functional Specifications, hands-on training objectives, and integrated simulator requirements from other-experiment EORs.

1.5.3 Simulator Top-Level Requirements Document

The Simulator Top-Level Requirements Document (STLRD) defines the overall methodology of each experiment simulator. It does this by tying together information set forth in the Simulator Approach Document (SAD), the Experiment Overview Report (EOR), and the Functional Specification. The SAD will supply the simulator skeleton, its major components and the strategy for their development. The Functional Specification will supply the simulator components defined by the SAD. Lastly, the EOR will provide a general experiment description, including data on both internal and external experiment interfaces. This information will be used to determine the required inputs and outputs for the various simulator functions. It is not intended that this document require a great deal of original effort, but rather that it be created largely by integration of the analytic products mentioned above. The major analytic responsibility in assembling this document is to translate the requirements from the Functional Specification onto the appropriate simulator components.

1.5.4 Experiment Simulator Requirements Document

At this point in the simulator development process, the major part of the analysis effort has been completed. The basic simulator approach has been determined and its various elements defined. Ideally, all experiment data necessary for simulator development has been identified and collected. The final step is to use this information to develop hardware and software implementation requirements in sufficient detail to allow simulator design and development efforts to proceed.

The ESRD organizes these requirements under the same simulator elements and sub-functions defined in the STLRD. Since the general simulation method for each sub-function of each element has been previously determined, all that is needed are descriptions of the specific requirements to accomplish each function. For software models, this consists of whatever is necessary to define its inputs, outputs, and behavior. For hardware components, this will mean system schematics, mechanical drawings, parts lists, and any other information about the actual

experiment needed by Design and Development (D&D) to create simulator hardware specifications.

1.6 Payload Training Systems Verification

The purpose of PTC Training Systems Verification is to provide NASA with systematic assurance that developed payload trainers will fulfill their role for PTC training in a correct, effective, safe, and economical fashion. The verification group, which is detached from the development group, is responsible for reviewing all delivered products with an objective and independent perspective to assess their technical adequacy. The verification group presents its findings at each of the major reviews.

The verification process involves a series of activities interfaced with the development process itself, and supports a more orderly and efficient implementation because each development phase produces a verified baseline for the next phase. As shown in the TRDS Template (Figure 1-1), verification activities begin during the Training Requirements Analysis phase and end with the Simulator Acceptance Review (SAR). As a result of the verification activities, errors are typically uncovered early in the development cycle before they have a chance to propagate. This early discovery promotes improved reliability, greater visibility, and reduced life-cycle costs.

The verification methodology as (1) the process of determining whether or not the products of each phase of the development cycle fulfill the requirements established during the previous phase and (2) the process of testing the simulator software to demonstrate that the software fulfills all functional requirements imposed by the requirements specification. To accomplish these goals, the verification process is organized into three major levels of verification activities:

- Increment-independent verification planning
- Specification verification
- Verification testing.

1.6.1 Increment-Independent Verification Planning

Prior to the development of the first SS increment training system, the verification group will produce a Generic Master Verification and Test Plan which will guide the verification process during the development of all the training systems. The Generic Master Verification and Test Plan will be a detailed expansion of the verification methodology described in Section 5.0 of this document.

1.6.2 Specification Verification

The purpose of Specification Verification is to allow in-progress verification of the training development process. The verification group examines both the simulator and non-simulator training development activities. Specification Verification creates a series of verified baselines upon which the instructions can be developed and tested. Specification Verification is an iterative process that occurs throughout the various phases of the training development cycle and includes the verification of: Training Objectives, Instructional Plans, Simulator Requirements, Designs, and Code Listings.

The purpose of verifying training objectives is to assess whether the Objectives Hierarchy for each experiment, as prepared by the responsible PI, are a fair representation of the training needs for that experiment. The purpose of Instructional Plan Verification is to determine whether the instructional media, with an emphasis on the computer-applicable portion of the Instructional Plan, represent a clear and accurate description of the training needs. The verification group analyzes Simulator Requirements to ascertain that the data systems requirements reflect the needs expressed in the Instructional Plan. During design verification, the verification is to allow a "code walk-through" of the code listings to determine whether the actual code implements the described designs.

1.6.3 Verification Testing

The purpose of verification testing is to plan and conduct tests to verify that the implemented trainers fulfill the simulator requirements. This testing does not include the testing responsibilities of the developer. Verification testing consists of three types of testing. Increment-independent simulation environment testing is performed to verify upgrades to the underlying simulation environment. Informal "free-form" testing is conducted to checkout the overall soundness and integrity of the simulator system.

Finally, acceptance testing is performed to execute the Acceptance Test Procedures in a controlled environment as defined in the Acceptance Test Plan. Verification testing is concluded with the Simulation Acceptance Review at which the testing results are presented and a selected subset of the test is repeated. The Payload Principal Investigator is encouraged to witness the formal tests, and to participate in any informal "free-form" testing as desired.

1.7 Validation

PTC Training Validation defines a process to ensure that the total training system developed for each Space Station experiment fulfills its overall training objectives. Unlike Verification, which is concerned with a simulator's individual capabilities, Validation is a process of evaluating a simulator's integrated ability to fulfill its purpose which is to provide training. In addition to simulator or hands-on media training, the Validation process involves evaluation of the academic training which will be provided as part of the total training offered for each experiment. Verification and Validation, which use the same tools and analyze the same data items, have been described elsewhere as intertwined activities throughout the development process. For our purposes however, Validation will be a separate activity starting later in the development process when the parts have been integrated and the final product is to be evaluated.

Validation will be performed by either the same people who are performing Verification, or at least by a group detached from the development crew. This Validation group, known as the Validator provides NASA with an objective and independent perspective to assess the training system capability to meet its objectives.

Training systems should be validated by comparing them with the training objectives and functional requirements from which they were designed. These criteria are 1) one step removed from the specific implementation details which were the focus of Verification and 2) relate directly to the various training functions of the system.

The Validation procedure therefore, will consider all stages of training from familiarization to integrated mission simulations. For example, the academic training objectives will be used to validate CBT courseware and classroom lessons, while hands-on media Functional Specifications will be applied to simulator training validation. The Validation process will consider a wide variety of inputs, such as JSC concerns, PI-provided training objectives, and integrated training functions which were factored into the Functional Specification before it was finalized.

The Validation process begins with the production of Test Plans which will be performed to validate all training development end-products. A Test Plan is defined as a set of directions for conducting a test which state conditions, methods, and procedures to be used. As shown in the TRDS Template (Figure 1-1), Test Plan development for academic instruction begins about midway through the detailed design phase. Actually, it could start as soon as the appropriate Academic Lesson Specifications have been verified. The Lesson Specifications define the lessons to be

produced, and so are necessary as guides for Test Plan formulation in lieu of the actual lessons, though they are not directly used as Validation criteria.

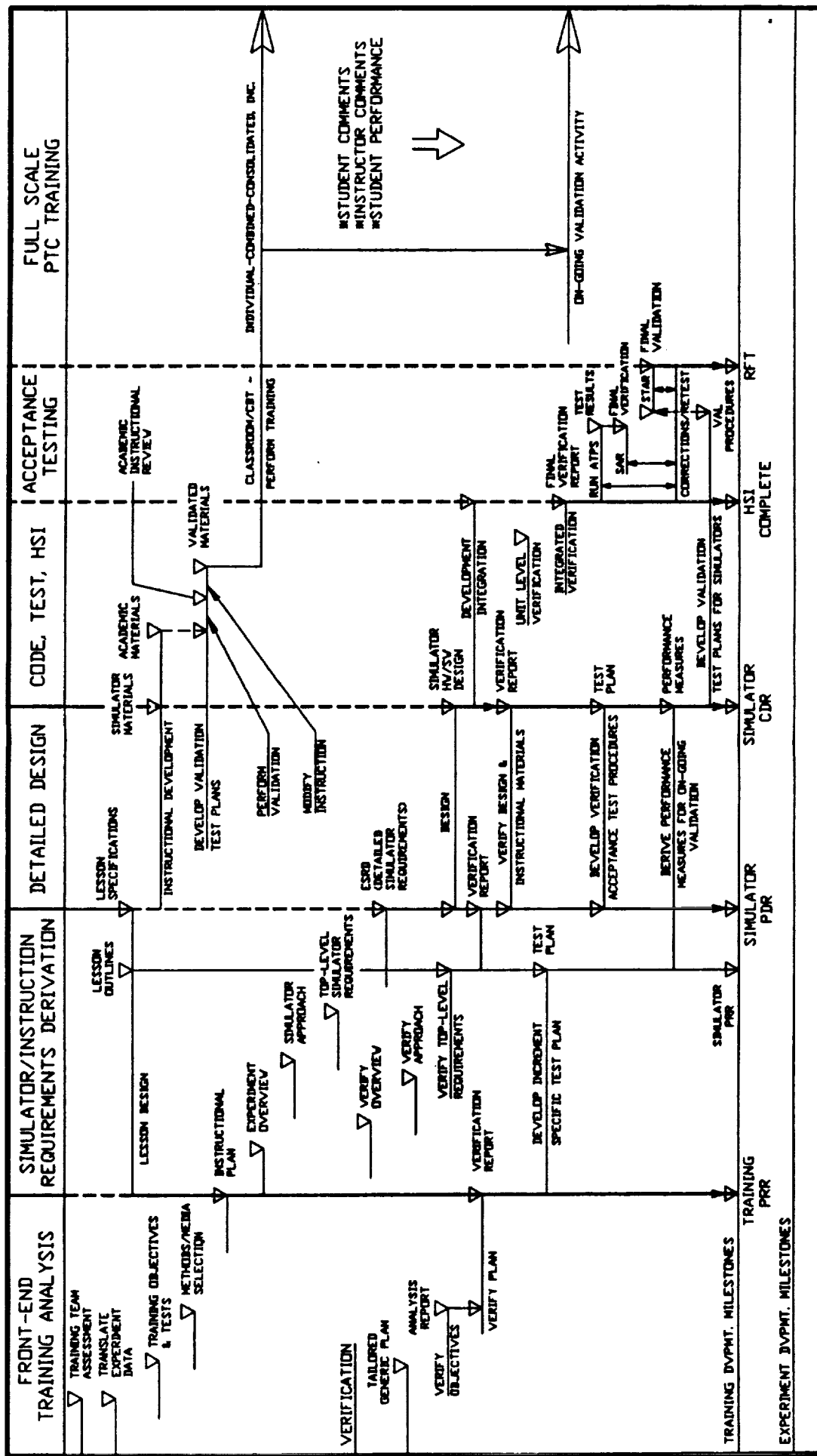
Test Plan development for hands-on or simulator instruction begins after simulator CDR, when instructional materials supportive of simulator training become available. Like academic Test Plan development, this effort could start sooner. It could begin as soon as finalized hands-on media Functional specifications or hands-on media Lesson Specifications have been approved. The Functional Specifications define the simulator functionality necessary to meet allocated training objectives. The hands-on Lesson Specifications define the supporting lessons and instructional materials which will be used in conjunction with the simulator to provide hands-on training.

Test Plans will be used to validate each simulator, each lesson, and to evaluate the overall integrity of the provided training system. Validation of Academic Instruction will commence as soon as the academic lessons, courseware, and supportive materials are complete, but before classroom or CBT training is scheduled to start. Validation procedures for hands-on training will be conducted for each simulator at its Simulator Training Acceptance Review (STAR). See the TRDS Program Template (Figure 1-1) for a graphical representation of this scheduling.

Once a training system has been validated, and pronounced Ready For Training, further validation activities will continue throughout the training cycle. Ongoing Validation will evaluate student performance in various ways to ensure that effective training occurs, and to detect and diagnose problems with the hardware or with the training regime. Corrective changes will be recommended both for current training, and for the training development methodology.

1.7.1 Academic Instructional Validation

This is where the lessons and instructional materials designed to fulfill academic training objectives are validated in actual use with academic media such as classrooms or CBT terminals. Because Verification will have been performed on the Lesson Specifications from which these academic end-products were designed, validation testing will ensure that the various instructional elements in combination, will meet their parent training objectives. Since the training objectives were derived from the tasks to be performed by different crew members, their use as validation criteria will ensure that the different training needs of the various flight and ground crew will be met by the proposed curriculum.



30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 09

FIG. 1-1 TRAINING PROGRAM DEVELOPMENT TEMPLATE

1.7.2 Hands-On Media Validation (Including Simulators)

Hands-On Media Validation is the process of ensuring that the various elements which have been developed for hands-on training provide the proper functionality to support all training objectives and planned use. These elements are comprised of simulator hardware and software, support equipment, training scripts, lesson plans, and any other aids required to facilitate hands-on training. In contrast to Verification, which tests instructional materials and simulator hardware and software for their individual characteristics, Validation will ensure that all of the elements work in combination to provide the required training. The hands-on media Functional Specification for the training simulator and higher level hands-on training objectives will be the primary criteria for hands-on training Validation. The Specification was developed from hands-on training objectives, which in turn were derived from the tasks performed by different flight or ground crew members. Therefore, like the academic training objectives used for validation of academic instruction, the use of the Functional Specification and hands-on training objectives as validation criteria for hands-on instruction will ensure that the different training needs of the various flight and ground crew will be met by the simulator functionality.

1.7.3 Ongoing Validation

After determining (through Validation) that the correct training systems have been designed and built, it is desirable to validate on a continual basis that the training systems are providing correct training. This will afford a degree of quality control for the immediate training process as well as to generate recommendations for improvement of the training development system for future training. Rather than focusing on training design criteria, as does the initial validation, Ongoing Validation will detect problems by evaluating student performance.

1.8 Training Program Template

Figure 1-1 depicts a top level flow of training development activities laid out along a launch-oriented timeline. Development activities up to the start of training are confined to within a 15 month window, with follow-on maintenance and training activities extending through the operation life of each payload. Although it is not shown on the chart, the chart assumes 12 months for PTC training (including classroom and CBT) followed by six months of training at the Space Station Training Facility (SSTF).

Roughly four months are allocated for front-end requirements definition activities, followed by three months to analyze simulator requirements, six months for detailed design and development, and two final months for acceptance testing. Verification activities will be conducted on an ongoing basis from Training Needs Assessment through Validation testing. Validation is performed once as the conclusion to Acceptance Testing and on an ongoing basis throughout the experiment training system lifetime.

2.0 TRAINING NEEDS ASSESSMENT METHODOLOGY

Training Needs Assessment is the process of defining the training which must be performed in order to prepare flight and ground crews for Space Station payload operations. The training defined will encompass all stages from introductory experiment familiarization, to experiment operation, up to integrated operations with other Space Station facilities. The scope of training is for all ground and flight operations necessary to accomplish the mission and science objectives of the Space Station Freedom experiments.

Needs Assessment begins with an organization of experiment and programmatic data into a format suitable for training development, traceability, and configuration control. This data is then analyzed to determine the tasks to be trained. The tasks are finally translated into training objectives and tests which will define what the students must learn in order to operate the experiments successfully.

2.1 Analysis of System Requirements

The first step in the development of an experiment training system is to determine exactly what is to be trained. This is accomplished through an analysis of the experiment to identify tasks which the ground and flight crews must perform for the operation, maintenance, control, and support of the experiment during an increment. The information for this analysis is drawn from available experiment data, and its use is guided by NASA and PTC training policies and guidelines. Before training analysis begins, however, this information will be organized into specific data items and established in an Experiment Database. The format for these data items should be designed to facilitate their use in the training development methodologies. As new input data becomes available, it is entered into the established databases to maintain firm traceability between experiment requirements and characteristics of the developing instructional system.

2.1.1 Experiment Database

<p>ORGANIZE EXPERIMENT AND PROGRAMMATIC INFORMATION INTO AN EXPERIMENT DATABASE</p>

By collecting experiment requirements into an organized database, the training analyst will develop an in-depth understanding of experiment functions and interfaces. This database will be maintained as the source for traceability from experiment to training requirements. It should provide a description of the experiment in terms of:

- (a) Mission or purpose
- (b) Functions or performance required to satisfy experiment objectives
- (c) Major subsystems and components used to structure the experiment
- (d) Equipment or materials required to support the experiment
- (e) Established concepts, policies or procedures for experiment operation, maintenance or use
- (f) The functional responsibilities of the people who will operate, maintain or use the experiment

Figure 2-1 illustrates data items which are either necessary or helpful to payload training development. Information to complete these items will be solicited from the PI, developed by training personnel from PI inputs, or provided by the Space Station Freedom Training Program.

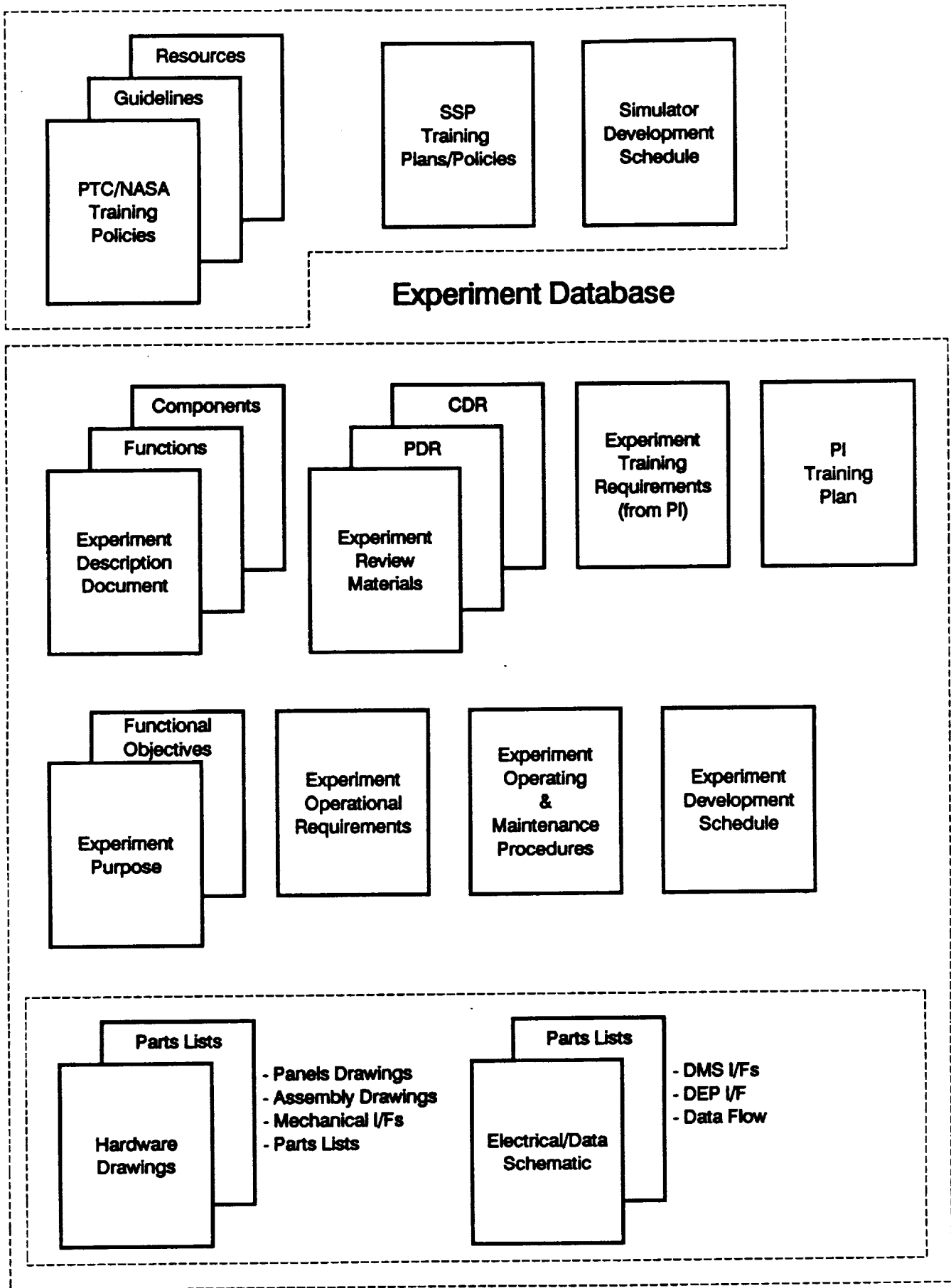


Figure 2-1. Experiment Database

One aspect of the training agreement with each PI should be to provide him or her a detailed definition of the data required about his or her experiment for training development and the preferred format for that data. For example, one product of training analysis will be an Experiment Description Document for each experiment. A "fill in the blanks" template of this document (possibly furnished in a word processor "merge" mode) could be supplied, so that the PI would be able to provide experiment information in exactly the form required. Time would be saved, even if later the inputs have to be rewritten since a template would give the PI a much better understanding of the actual data requirements. This will help to standardize inputs to the TRDS, maximize the accuracy and quantity of the inputs, and minimize subsequent training analysis efforts. It is recognized, however, that some PIs will be unable to provide all the information in the requested format and at the required time. Early efforts should therefore be made to size and scope the required development effort with respect to anticipated data availability.

As information on each experiment becomes available, it is entered into the configuration controlled Experiment Database, so that traceability may be established between experiment requirements and simulation and training requirements. The more compatible the incoming information is with the database structure, the easier this process will be. "Database" in this context implies, but does not mandate a computerized utility. Many documents may be left in hardcopy or magnetic media, however, they must be maintained and configuration controlled. This database will be drawn upon to derive a detailed hierarchy of tasks (and associated attributes) necessary for experiment maintenance and operations.

Experiment Description Document: This includes the experiment top level functions, components, interfaces, and principles of operation. If initially produced by the PI in accordance with specific TRDS guidelines, it would aid the training analyst in basic understanding of the experiment. In addition, if a document template were provided, this information could immediately provide the basis for an experiment description document deliverable (Experiment Operating Report [EOR]).

Experiment Purpose: The PI should provide a clear, unambiguous explanation of the purpose, and functional objectives of his experiment. This will help in developing Job Performance Requirements and training Requirements.

Drawings, Schematics and Associated Lists: These are the electrical, mechanical, and data schematics, and the associated parts lists generated by the PI/PED (Principal Investigator/Payload Element Developer) during the process of

experiment design and development. Though in many cases the production drawings will not be available in a timely manner for simulator development and will be prone to frequent revision, even preliminary and "in-progress" versions will be valuable for providing insight to experiment methods, data flow, interfaces, and for deriving inputs to simulator hardware design.

Experiment Training Requirements: Since the PI is best acquainted with his or her experiment's purpose, the PI can be expected to provide insight into its most important operational aspects and hence, the most critical tasks for training. The training analyst will augment or modify these PI-provided requirements with those resulting from his or her own research to arrive at a complete list for training development.

Experiment Operational Requirements: This describes all of the resources needed for experiment operation such as data, physical support, sensory inputs etc. Includes operator roles, identities, and functional responsibilities. This information will be used to help develop Job Performance Requirements, simulator approach, and lesson plans.

Experiment Operational and Maintenance Procedures: These would comprise a direct input to lesson plans, training scripts etc. as well as providing understanding of tasks and task criticality.

Experiment Development Schedule: Close monitoring of experiment milestones will aid in planning for training development -- especially with respect to strategies to compensate for anticipated data inadequacies.

PI Training Plan: In order to conduct efficient training, it is necessary understand what the trainees already know, as well as what they need to know. The PI Training Plan should describe the experiment training which will be provided at the PI sites prior to training at the PTC. From this, the abilities, skills, and knowledge which the flight crew will possess upon entry to the PTC may be determined. The necessary instruction then, will be determined as the difference between the final training objectives and what the training has already accomplished.

SSP Training Plans: This is information pertaining to the instruction the flight crew and ground crews will receive (before PTC training) on SS systems, POIC systems, or any other systems used during payload-related activities. This will be used to determine the amount of incidental and explicit training on those systems which the PTC would have to provide to enable payload training scenarios.

Experiment Review Materials: Materials presented at experiment development reviews such as PDRs and CDRs should be obtained for

general information as well as to gauge experiment progress and provide early guidance to simulator approach definition.

NASA and PTC Training Policies: These include information on the training resources available, the prevailing training philosophy, guidelines as to the degree of training to be provided, amount of crosstraining, options for OJT, job performance aids etc. Training for every experiment should be developed under uniform, consistent, and well-understood programmatic guidelines, so that the training produced will accurately reflect overall SSF training goals.

Simulator Development Schedule: A development strategy (documented by the Simulator Development Schedule) should be developed for each experiment from the very beginning of requirements analysis, based on experiment progress, anticipated data availabilities, and programmatic factors. This will provide an early "heads-up" for potential problems and allow early resource planning.

Trainee Information: This includes resumes and profiles of the individual trainees slated for each increment. This information will be used to develop training regimen for each trainee, customized for the skills and knowledge which they already possess.

2.2 Analysis of Training Requirements

Once a body of knowledge has been organized about the operation of an experiment, this knowledge may be analyzed to derive the tasks necessary to operate and support the experiment during a Space Station increment. These tasks may be organized into a Task Hierarchy consisting of Activities, Phases, Tasks, and Sub-Tasks. Though this nomenclature divides the Hierarchy into different levels in order to define superordinate and subordinate relationships, it should be noted that all hierarchy elements may still be generically referred to as tasks.

After a Task Hierarchy has been established, the tasks are characterized and classified in various ways. Task attributes such as Conditions and Standards of Performance are added to them, as well as a number of other properties such as Criticality, and Difficulty. When each task has been sufficiently detailed, an Objective Hierarchy is derived from the Task Hierarchy, representing the behaviors which are to be trained in order to accomplish the tasks. Finally, Criterion Tests and Diagnostic Tests are derived for each Training Objective.

2.2.1 Construction of a Task Hierarchy

LIST OF ALL MAJOR ACTIVITIES WHICH SUPPORT THE OPERATIONS OF A PARTICULAR EXPERIMENT

The following criteria shall be considered when determining major activities. An activity:

- a) Has a set of operations usually performed by a system of individuals.
- b) Has a clearly definable beginning and end point. Not all task listings will have multiple distinct activities.
- c) Is often identified with an end goal of coordinated crew activity.

Examples might be "Conduct Experiment XYZ Research" or "Conduct Emergency Experiment XYZ Operations".

SELECT AN ACTIVITY AND DIVIDE IT INTO PHASES

The following characteristics shall be considered when identifying phases:

- a) It can be given a name.
- b) It has a logical beginning and end point.
- c) It occupies an exclusive time slice.
- d) All phases taken together describe the entire activity.

Examples might be "Pre-Installation", "Experiment Operation", or "Post-experiment."

WALK THROUGH EACH PHASE, LISTING ALL TASKS

Tasks are named for the products they create or the processes they use. Phases are named for the time periods they occupy. They may be distinguished on that basis. The following characteristics shall be considered when identifying tasks:

- a) It is a significant operator activity (with a name).
- b) It has an observable beginning and end point or results in a consistent product.
- c) It usually includes a consistent sequence of specific behaviors (sometimes called "subtasks").

Examples could be "Perform Experiment Checkout," "Activated Experiment," or "Align Crystals for Maximum Emissivity."

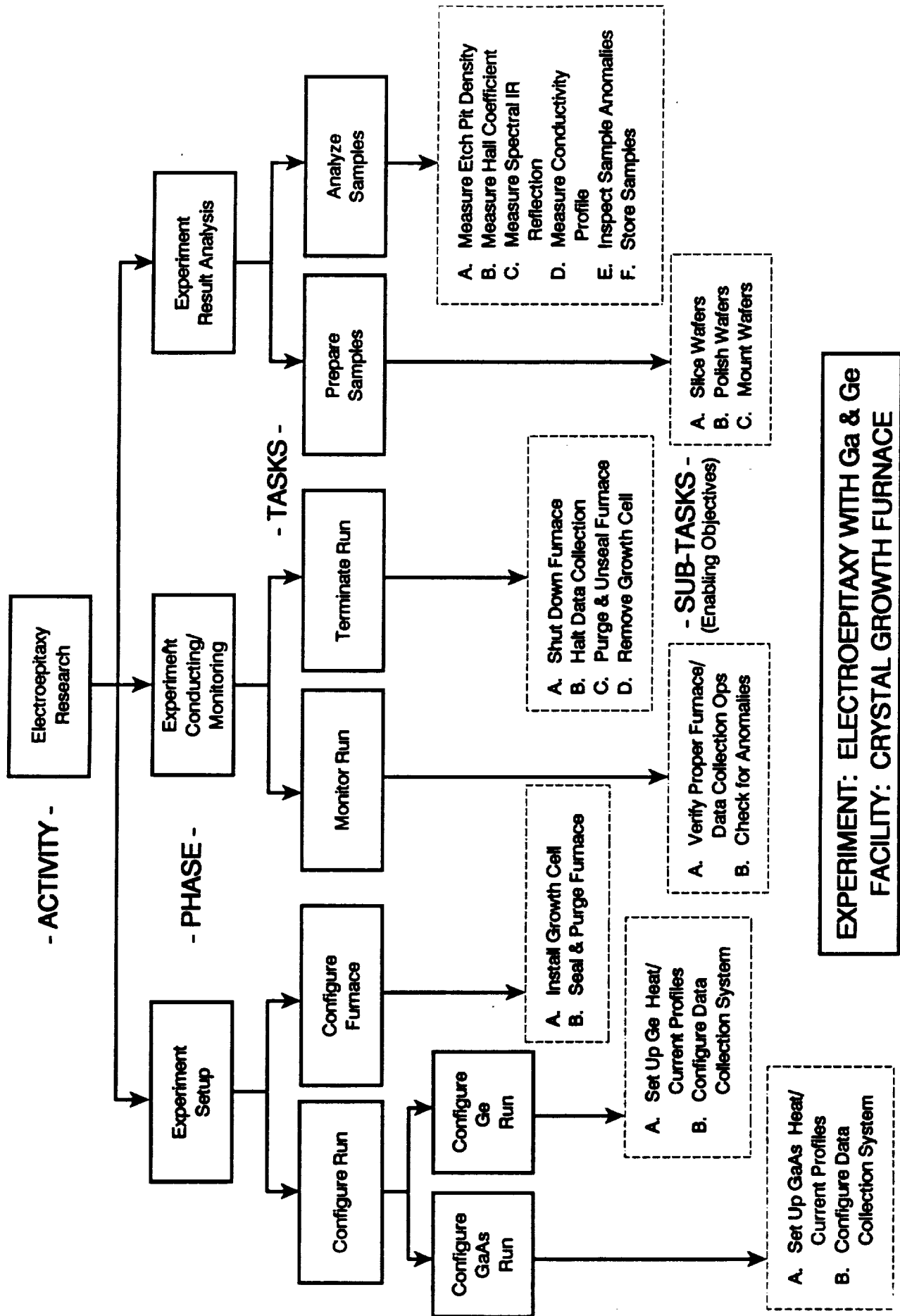
In decomposing tasks, care should be taken to break out a sufficient number of discrete tasks to enable a clear understanding of experiment procedures. Without sufficient detailing, important tasks may be omitted from training or assigned to an incorrect level or location in the hierarchy. On the other hand, intermediate, or component skills and knowledge which appropriately should be added during development of objective hierarchies should not be included. Generally, the appropriate level of detail can be determined as follows:

- a) The point beyond which task components, rather than whole tasks will be entered.
- b) The lowest level at which performance will be evaluated independent of other contiguous tasks.

One way to develop a task hierarchy for payload training is to organize it around the experiment facilities. For example, a logical task hierarchy for operations concerning the SSF Furnace Facility would be a breakdown of all the tasks required to operate this facility in normal and contingency modes. Each individual experiment using the facility would have as its own task hierarchy a subset or modification of the overall task hierarchy for Furnace operation. If, on the other hand, rather than utilizing a payload facility, an experiment utilized its own process equipment (facility) in a stand-alone rack, this method could still be used. For an experiment such as Quantized Vortices in Super-fluid Helium for example, the tasks would be organized simply around operation of the experiment facility equipment, which for that experiment is housed in a dedicated rack.

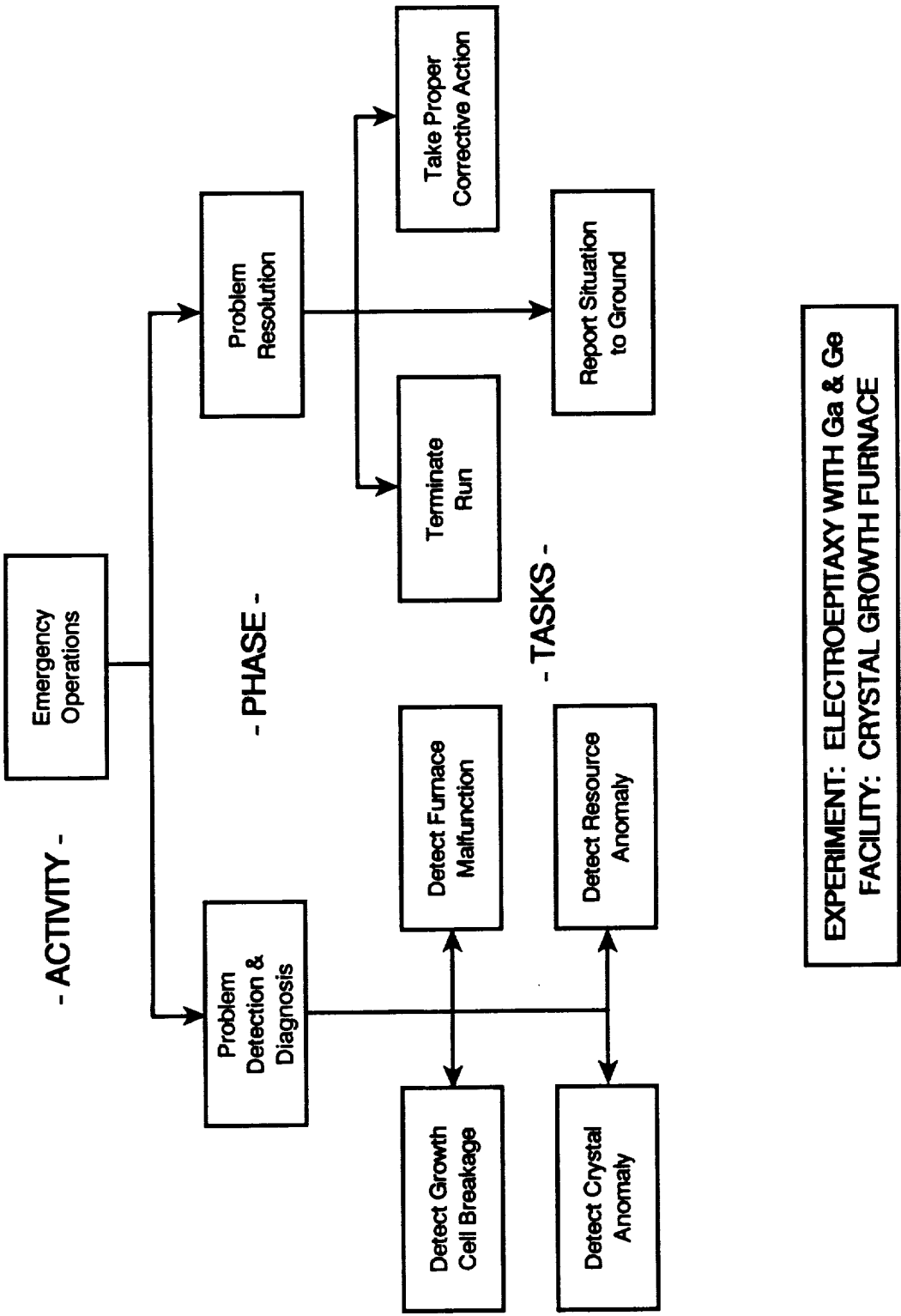
This is a good method for payload training organization, because it allows complete training system development without having to be concerned about which crew member position (Payload Specialist, Mission Specialist, etc.) is responsible for specific duties. The training facilities can be developed to train all necessary tasks, and trainees can assigned for training according to whatever division of responsibilities is currently in effect.

An example of a hierarchy organized in this manner is shown in Figure 2-2(a) and (b). The Task Hierarchy shown represents a modification (perhaps very minor) of the Task Hierarchy developed for general operation of the Crystal Growth Furnace in which the experiment will be conducted. The approach here is to develop a baseline Task Hierarchy for operation of the experiment facility, and then modify or supplement it as necessary for each experiment using the facility. Whereas in the example given, the experiment appears to be simply a direct use of the Crystal Growth Furnace; there are other experiments which will contain their own control systems and processes, and yet will still be interfaced to a "host" experiment facility. In those cases, the Task Hierarchy for the experiment will likely be an addendum to the facility's Task Hierarchy. In any case, the objective is to not re-invent training which has already been assimilated, but to build on what has already been accomplished.



**EXPERIMENT: ELECTROEPITAXY WITH Ga & Ge
FACILITY: CRYSTAL GROWTH FURNACE**

Figure 2-2(a). Sample Task Hierarchy (Also see Sample Task Listing, next page)



EXPERIMENT: ELECTROEPITAXY WITH Ga & Ge
 FACILITY: CRYSTAL GROWTH FURNACE

Figure 2-2(b). Sample Task Hierarchy (cont.) (Also see Sample Task Sheet, next page)

Note that the tasks in Figure 2-2(a) have been broken down to a degree which approximates the guidelines given in a) and b) above. How to break down a task into reasonable components at this stage is not "cut and dried" but remains a judgmental issue. The sub-tasks, or Enabling Objectives shown in the dotted boxes are not part of the Task Hierarchy, but have been included for continuity with the Objectives Hierarchy.

While an attempt has been made to clarify development of the Task hierarchies, it should be fairly obvious that within the given guidelines, many different structures could be derived from the same input data. Since the initial organization may have a significant impact on the instructional configuration which results, it is suggested that the developer be guided by the experiment's purpose. In other words, try to organize the hierarchy of tasks in a way which will emphasize the tasks which most strongly support the perceived experimental objectives. Once the hierarchy structure has been defined, the tasks should be numbered according to a system which will reflect their subordinate and superordinate relationships.

LIST ALL ADDITIONAL TASKS REQUIRED TO PERFORM
UNDER EXTRAORDINARY CONDITIONS

As a final step in the process of determining all tasks, each activity, phase, and task should be re-examined to determine if there are any situations under which it would be performed differently. This would include emergency situations where personnel or experiment objectives would be threatened. It would also include abnormal situations such as unexpected test results which could entail procedural or experiment configuration changes. Any new activities, phases, or tasks discovered in this manner should be incorporated as appropriate, into the task breakdown structures.

2.2.2 Assignment of Task Attributes

SPECIFY CONDITIONS AND STANDARDS OF PERFORMANCE
FOR EACH TASK, AS APPROPRIATE

After all activities, phases, and tasks have been considered for a given experiment, the Conditions and Standards of Performance associated with each are specified. A Standard of Performance is defined as a measure of the minimum proficiency with which a task can be accomplished. It is usually defined in terms of a

parameter which can be quantified, such as speed, accuracy, or time. Examples could be "The measurement should be within +/- 3.0 degrees of actual arc;" "Assembly should be accomplished without error, and within five minutes;" or "Measurements shall commence within one hour of flare discovery and must include three peak readings." The conditions under which a task must be accomplished are usually more general in nature and concern the work environment, tools or job aids used, location of task, event which initiates task, etc. Examples include sensory conditions, availability of checklists and tasks which must be concurrently performed.

EXAMPLE, TASK STATEMENT:

"At the end of each XYZ experiment run, the Payload Scientist deactivates the XYZ collator at the MPAC using the normal shutdown procedure.

Task: Deactivate the XYZ collator.

Condition: At the end of each experiment run, using the normal shutdown procedure, and an MPAC.

Standard: "Correctly" is implied.

These informational additions are made as appropriate, at each level in the hierarchy. Some activities in each level will have such attributes, and some will not. For example, the "operate experiment" phase may have an overall requirement to "perform 10 different heat and current profiles in one experiment run". This requirement, while not directly impacting training on that level, will probably result in time limitations being imposed for the completion of experiment tasks at lower levels in the hierarchy.

A Task Hierarchy is comprised of the tasks which must be accomplished, the conditions under which the tasks must be executed (why, when, where, and with what), and their required standards of performance. Once developed, Task Hierarchies will be accessed throughout the remainder of training development to help derive experiment instructional objectives and as a data source for detailed simulator and academic requirements.

**ASSIGN ADDITIONAL CHARACTERISTICS AND
ATTRIBUTES AS APPROPRIATE, TO EACH TASK**

In addition to Conditions, and Standards of Performance, each task will include a number of attributes which will fully define

the task for training analysis purposes. These attributes (as applicable to each task) include:

- a) Extent of Previous PI of SSP Provided Training.
- b) Number (and identity) of People Who Will Perform the Task.
- c) Criticality of the Task. Criticality refers to the task's relative importance to mission success as compared to the importance of other tasks.
- d) Frequency of Performance.
- e) Learning Difficulty.
- f) Time Interval Before First Performance.
- g) Personnel Safety Considerations.
- h) Tools and Equipment Needed to Perform Task.
- i) Time Required to Perform Task (Minimum and/or Maximum).
- j) Training Classification (with Rationale)
- k) Cross-reference to same task under other task groupings.

Typically, the greatest level of detail for task instructional attributes, Conditions and Standards of Performance is found at the lowest levels of break down. Therefore, attributes may be most easily assigned by starting at the bottom of the hierarchy, and collecting them upward as appropriate.

2.2.3 Task Classification

CLASSIFY EACH TASK FOR TRAINING ON THE BASIS
OF ITS INSTRUCTIONAL ATTRIBUTES

The first use for the task attributes will be to make an initial classification of each task as regards its need for training. The results of this classification process will be recorded as an attribute (j above), along with a rationale for the classification. The tasks may be placed in one or more of the following categories:

- a) General training: Tasks which are above entry-level skills and knowledge but which are performed for more than

one experiment. Examples of this could be an experiment de-installation procedure or enabling dedicated experiment data lines.

b) Mission qualification training: Tasks that are specific to a particular experiment.

c) Refresher or Proficiency Training: Critical or frequently performed tasks which may need refresher training, as well as tasks which will not be performed until well after they are trained in the normal curriculum.

d) Continuation training: Tasks which are critical, or which by their nature require repetitive training to maintain ability. Tasks involving hand and eye coordination or other motor skills could fall into this category.

e) No training: Tasks may be deselected for training if they are trivial, rarely performed, or are part of the entry level skills of the ground and flight personnel. A Task may also be excluded if it is adequately trained in another part of the curriculum. Care of course should be taken to not exclude tasks which though previously trained may require refresher or proficiency training. Even if a task is excluded from training at this point, it should be maintained in the task listings if that mission requirements or entry-level skills change.

While this initial task classification is tentative, it is important to record a rationale for every decision made. With reasons documented for every decision, training program requirements such as these can easily be updated as more experiment information becomes available.

Once preliminary Task classification and screening has been performed, the developer will document each Task and its attributes on Task Data Forms (Figure 2-3). The set of Task Data Forms should include missing Tasks or Tasks for which information is incomplete, as well as all of the established information. This set of forms will most likely be produced automatically for the developer through software utility. The developer will use the forms as a means of communication with the PI in resolving data discrepancies. At an appropriate time, the developer forwards the entire set of completed Task Data Forms to the responsible PI as part of experiment training Verification.

EXPERIMENT: Electroepitaxy with Ga & Ge			
TASK: Prepare samples		TASK DESIGNATOR: 1:3:1	
SUBORDINATE ACTIVITIES: Slice wafers 1:3:1:1 Polish wafers 1:3:1:2 Mount wafers 1:3:1:3		EXPERIMENT REQUIREMENT REFERENCE: ERD #XXXX Electroepitaxy Experiment, Page X, Paragraph Y	
JOB FUNCTION(S): Experiment Operator			
PARENT TASK: Experiment Results Analysis 1:3			
RESPONSIBLE CREW POSITION(S): Payload Specialist, Mission Specialist			
HUMAN INTERFACES:			
ACTIVITY CODE: 01			
TASK TRAINING CLASSIFICATION: General Training			
CLASSIFICATION RATIONALE: This task is common to several experiments			
TASK SPECIFIC TRAINING FACTORS			
Performance Frequency:	3	(1-6)	Time to Perform Task: 1 hr
Task Criticality:	5	(1-6)	Interval Before First
Learning Difficulty:	3	(1-6)	Performance: 2 mo.
TASK DESCRIPTION			
Action and Item Acted Upon:		Operator removes crystal from growth cell and prepares crystal for study by slicing it into wafers, polishing wafers to varying degrees, and mounting them on sample trays.	
Task Constraints, Contingencies:			
Support Equipment, Materials, Tools, References:		Materials Handling Glovebox, crystal cutter, crystal polisher/grinder, Experiment Specifications Notebook, and samples trays	
Consequence of Inadequate Performance:		Insufficient number of sample wafers for analysis	
Hazard Potential:		Danger to hands from crystal cutter	
Controls:			
Displays:			
Inputs (Action Determinants):		Used growth cell tagged for on-orbit analysis	
Outputs (Standards of Performance):		The required number of unbroken wafers must be within +/-3 mm of specified thickness and 10% of specified smoothness.	
Commonality:		Most electroepitaxy and directional solidification experiment will require this task	
NOTES:			
Payload Specialist has primary responsibility; Mission Specialist has secondary responsibility for this task.			

Figure 2-3. Sample Task Sheet

2.2.4 Training Objective and Test Development

Once the Task Hierarchies for an experiment are defined, they can be used to develop Objective Hierarchies. These comprise Training Objectives, Criterion Objective Tests, and Diagnostic Tests. In contrast to the Task Hierarchy which states what must be done to operate the experiment, the Objective Hierarchy describes what must be learned in order to perform experiment tasks. The training objectives will be used as the framework for all instruction, both academic, and hands-on, with simulators. Lessons will be designed around accomplishment of the objectives and simulators will be designed with the functionality and fidelity necessary to train the specific objective behaviors. Objectives will be used to determine lesson sequence and aid in training media selection.

Criterion Tests and Diagnostic Tests will be derived for each objective to evaluate students' accomplishment of the training system. As a check, developed objectives will be compared against objectives previously identified by the PI to spot omissions and contradictions (if any). Once the Training Objectives have been specified, media allocations may be made based upon them and upon the previously developed task attributes. The results of Training Objective and Test Development is a hierarchy of Objectives with related Test Items. These Objectives and Tests identify what is to be taught, and how the results of this teaching will be demonstrated.

It is anticipated that much of the training to be performed at the PTC will be "learner-controlled". Learner-controlled means that the students, who are highly motivated, are free to use any or all of the training resources as they deem necessary. For these cases, the students could be given the set of Training Objectives along with the appropriate Test Items, so that they can determine for themselves when they have reached their objective.

Construction of a Training Objective Hierarchy: In Task Analysis, the tasks which must be performed in order to accomplish experiment objectives were identified and characterized. In Training Objective Development, the skills and knowledge necessary to accomplish these tasks are stated behaviorally. That is, objectives are stated in terms of desired student behavior. A Training Objective is a precise statement that specifies what a student must do to demonstrate that the desired learning has taken place. It includes the minimum standard of performance proficiency expected (which may be perfection) and the conditions under which this behavior is to be shown.

EXAMPLE, TRAINING OBJECTIVE:

Given the use of an MPAC, and a standard mass spectrometer, calibrate the XYZ sensor to within 3% of the primary wavelength under observation.

<u>BEHAVIOR or PERFORMANCE</u>	<u>CONDITIONS</u>	<u>STANDARDS</u>
Calibrate the XYZ sensor,	using a standard mass spectrometer, from the MPAC.	Calibrate to within 3% of the primary wavelength.

To be effective, an Objective will state clearly what the student must do to demonstrate learning, preferably using action verbs. This may or may not be the same as the Task Statement. An Objective statement such as "the student will understand how to activate the XYZ sensor" is poor because it is open to wide interpretation. The Objective must specify some observable behavior such as: "the student will write the steps necessary to activate the XYZ sensor, in the correct sequence". This allows all training personnel to understand exactly what is to be learned and how the student is to demonstrate learning.

DEVELOP TERMINAL TRAINING OBJECTIVES FROM THE
TASKS SELECTED FOR TRAINING IN THE TASK
HIERARCHY

The initial body of Terminal Objectives will be drawn from the tasks-to-be-trained in the Task Hierarchy. A Terminal Objective is one which reflects the accomplishment of an identified job task. It usually demonstrates the acquisition of a combination of skills and knowledge. The "Behavior of Performance" statement of the Objective may often be taken directly or with minor rewording from each Task Statement. Likewise, the Conditions and Standards of Performance for each Objective may be derived from these previously established sources.

There is not necessarily a one-to-one relationship between Objectives and Task statements in the Objective Hierarchy. Several Objectives may support a task in one instance, while one Objective might support several tasks in another. In addition, the focus of the objectives will be to specify what must be learned, rather than what must be done, for experiment operation. Therefore, some tasks may not be appropriate as objectives, or may be performed in a different sequence for learning, than they are operationally.

Also, the Conditions and Standards of Performance for the operational environment may have to be altered to be appropriate for training. Safety considerations or those of practicality may dictate changes in the way a task may be taught. Certain operations might be repeated for training emphasis, or condensed in order to fit within a reasonable training session.

Since there is not a one-to-one relationship between objectives and tasks, the training analyst is free to develop the Objectives Hierarchy in a manner which provides the most efficient training. In any case, the Task Hierarchy will be left intact to provide an audit trail back to the Experiment Database and as a data source for lesson development.

DEVELOP ENABLING OBJECTIVES AS NECESSARY FOR
APPROPRIATE TERMINAL OBJECTIVES

After the Terminal Objectives have been defined, it may be necessary to break them down for training purposes into their component skills and knowledge. These components are known as Enabling Objectives which represent the intermediate skills and knowledge necessary to attain the Terminal Objective. Subobjectives in turn, may be derived from the Enabling Objectives until the most basic skills and knowledge necessary to be trained have been identified. For example, a Task such as aligning an experiment sensor antenna might require the following skills and knowledge:

<u>Skill of Knowledge Code</u>	<u>Skills and Knowledge</u>
100235461	Experiment Location
100567843	Operation XYZ Tuner from MPAC
100235469	Knowledge of Tuning Procedure
100235463	Ability to Interpret XYZ Sensor
100549276	Operation of SS Communication Network
100549279	Knowledge of SS Communication Protocols
100235467	Purpose of Tuning Procedure

When the Instructional Program is planned, the lesson designers will start with these basic skills and knowledge and work their way up the hierarchy until the trainees have mastered the component skills needed and can accomplish the Terminal Objectives. Figures 2-4 (a-d) are examples of Objectives Worksheets which can be used to develop training objectives. The examples given continue the process illustrated by Figure 2-2a for "Prepare Samples." Figure 2-5 shows the Objectives Hierarchy resulting from a breakdown of "Prepare Samples."

BEHAVIOR/PERFORMANCE	CONDITION(S)	STANDARD(S)
Prepare crystal sample for analysis Task Designator 1:3:1	Using the Materials Handling Glovebox, crystal cutter, sample trays, crystal polisher/grinder, and Experiment Specification Notebook; given a used growth cell, tagged for on-orbit analysis in the Materials Sampling Bin.	The specified number of unbroken wafers must be obtained from crystal sample. Wafers must be to within +/-3 mm of specified thickness and 10% of specified smoothness.
TERMINAL OBJECTIVE 1:3:1		
Given a growth cell tagged for on-orbit analysis, prepare crystal samples using the Materials Handling Glovebox, crystal cutter, sample trays, crystal polisher/grinder, and Experiment Specifications Notebook. Wafers produced must be within +/-3 mm of specified thickness and 10% of specified smoothness. The requisite number of unbroken wafers must be obtained from crystal samples		
MEASURE FOR TRAINING EFFECTIVENESS		
Perform above objective		
COMPONENT SKILLS/KNOWLEDGE OR SUBTASK ENABLING OBJECTIVE		
Ref. Code XXXX Slice crystal wafers to specified thickness XXXX Polish wafers onto sample trays XXXX Mount wafers onto sample trays		

Figure 2-4(a). Objective Worksheet

BEHAVIOR/PERFORMANCE	CONDITION(S)	STANDARD(S)
Slice crystal sample into wafers Task Designator 1:3:1:1	Using the Materials Handling Glovebox, crystal cutter, and Experiment Specifications Notebook; given a used growth cell, tagged for on-orbit analysis in the Materials Sample Bin.	Resultant wafers must be within +/-3 mm of thickness. The requisite number of unbroken wafers must be obtained from crystal sample.
ENABLING OBJECTIVE 1:3:1:1		
Given a growth cell tagged for analysis, slice the sample contained therein in the Materials Handling Glovebox, using the crystal cutter. The wafers must be within +/-3 mm of the thickness and at least the number specified in the Experiment Specification Notebook.		
MEASURE FOR TRAINING EFFECTIVENESS		
Perform above objective		
COMPONENT SKILLS/KNOWLEDGE OR SUBTASK ENABLING OBJECTIVE		
Ref. Code XXXX Locate required thickness/required number of wafers. XXXXX Operate crystal slicer XXXXX Mount wafers onto sample trays		

Figure 2-4(b). Objective Worksheet

BEHAVIOR/PERFORMANCE	CONDITION(S)	STANDARD(S)
Locate required thickness and number of wafers. Task Designator 1:3:1:1:1	Given a certain experiment and using the Experiment Specifications Notebook	Without error and without assistance
ENABLING OBJECTIVE 1:3:1:1:1		
As below, but only for one experiment at a time.		
MEASURE FOR TRAINING EFFECTIVENESS		
Given the requirement for specified information (thickness and number) for five kinds of experiments and a copy of the Experiment Specifications Notebook, locate the needed information without error and without assistance.		
COMPONENT SKILLS/KNOWLEDGE OR SUBTASK ENABLING OBJECTIVE		
Ref. Code XXXX Use of Experiment Specifications Notebook		

Figure 2-4(c). Objective Worksheet

BEHAVIOR/PERFORMANCE	CONDITION(S)	STANDARD(S)
Operate crystal cutter Task Designator 1:3:1:1:2	Using the Materials Handling Glovebox and crystal cutter; given a used growth cell tagged for an on-orbit analysis in the Materials Sample Bin and the required thickness and number of wafers.	Resultant wafers must be at least the required number, unbroken and the thickness specified to within +/-3 mm.
ENABLING OBJECTIVE 1:3:1:1:2		
Given a growth cell tagged for on-orbit analysis in the Materials Sample Bin, slice the sample contained therein in the Materials Handling Glovebox by using the crystal cutter. The wafers must be within +/-3 mm of the specified thickness, unbroken, and at least the number of wafers specified.		
MEASURE FOR TRAINING EFFECTIVENESS		
None. The skills and knowledges necessary to attain this objective will be trained at the SSTF.		
COMPONENT SKILLS/KNOWLEDGE OR SUBTASK ENABLING OBJECTIVE		
Ref. Code XXXX Location of Materials Handling Glovebox XXXX Alignment of cutting jig XXXX Actuation of the crystal cutter		

Figure 2-4(d). Objective Worksheet

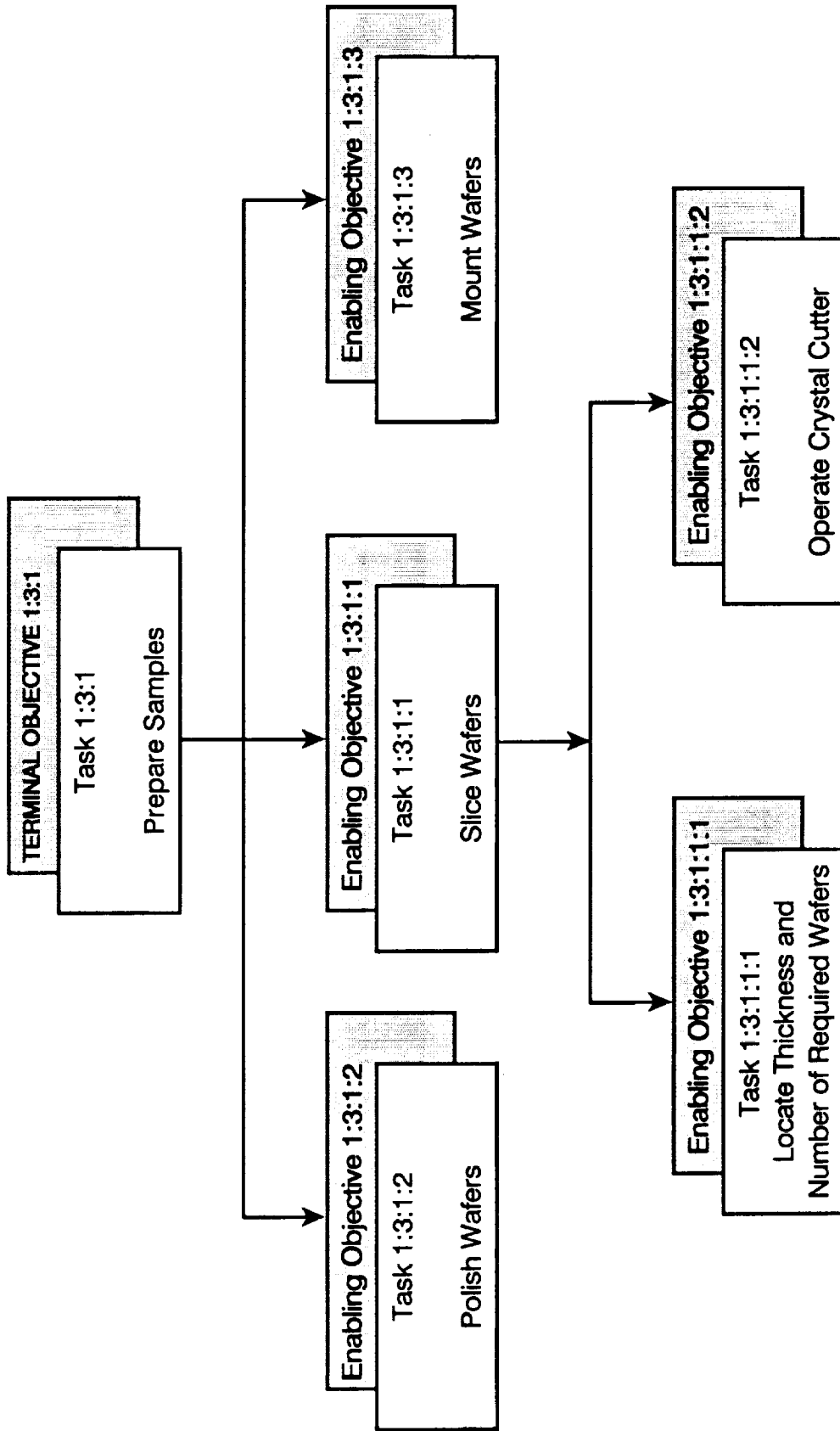


Figure 2-5. Breakdown from a Terminal Objective

ADD HIGHER LEVEL OBJECTIVES TO ADDRESS
ACTIVITIES ABOVE SINGLE EXPERIMENT OPERATION

Before an Objective Hierarchy is complete, it must be linked with the operation of other experiments represented by their own hierarchies. The mechanism for this linking is called simply a higher level objective. This type of objective is concerned with the development of skills and knowledge associated with the simultaneous operation of multiple experiments and with the accomplishment of Mission-level objectives. An example of this type of objective might be "Utililizing Ongoing Outputs from Experiment ABC, Conduct Experiment XYZ." The accomplishment of this type of objective (while building on the skills and knowledge for individual experiment operation) would be more concerned with skills related to integrated experiment operation such as teamwork skills, communication skills and timeline validation. Similarly, objectives such as "Conduct Whole US Lab Experiment Operations" or "Conduct Whole Station Experiment Operations" would concentrate on the resource juggling and coordinative skills necessary on those levels.

These objectives, rather than being drawn from the Task Hierarchies, would derive from Mission goals and objectives, and programmatic guidelines. Training scenarios to satisfy them would be designed into lesson plans from integrated experiment training within one Space Station Module, between modules, and between Space Station training facilities.

Once the structure of the Objective Hierarchy has been defined, the objectives should be numbered according to a system which will reflect their subordinate and superordinate relationships. Though it obviously cannot duplicate it, this should be related to the Task Hierarchy numbering system.

Criterion Tests:

DEVELOP CRITERION TESTS FOR EACH TERMINAL
OBJECTIVE IN AN EXPERIMENT'S OBJECTIVE HIERARCHY

For each Terminal Objective developed in an experiment's Objective Hierarchy, a Criterion Test will be developed. A Criterion Test is a measure of student performance based on an objective Standard rather than by comparing one student's performance against others. Development of these kinds of test are important, because they are used to measure the effectiveness

of the instructional system as well as to determine if students have attained the course objectives.

Equally important is to ensure that the test developed focuses on the achievement of the specified Criterion Objectives rather than other similar criteria. For example, an appropriate test item for the Objective "Measure the Thickness of a Vacuum-Sputtered Titanium Layer" would be to require the student to actually measure such a deposit. Writing an essay on the measurement techniques involved would not be an appropriate measure of the student's attainment of the objective. In general, the tests should require the same performance from the students that was required during the training.

The easiest way to focus testing on Training Objectives is to base the criterion test item solely on the requirements stated in the objective which it must measure. In many cases, the working in both the test item and the objective will be the same. For example, the objective "Using the XYZ Experiment Manipulator Togs and the ABC Thermal Probe, To Adjust the Boron Crystal to Within 2 Percent of Its Maximum Emissivity," may in itself be a good test. In other cases, some rephrasing may be necessary. In any case, the test should require the student to meet the same standards of performance. The test should also be conducted under the same conditions as specified in the objective.

These test items will be incorporated into Performance Evaluation Plans which will comprise a section in Lesson Specifications written during Syllabi Development (see Section 3.2.3). The Plan will be used during Validation to prove that the developed instruction satisfies the Training Objectives. After Final Validation, the test will be used in Ongoing Validation to evaluate instructional effectivity and to track student progress.

Diagnostic Tests:

DEVELOP DIAGNOSTIC TEST FOR EACH ENABLING
OBJECTIVE IN AN EXPERIMENT'S OBJECTIVE
HIERARCHY

Diagnostic Tests will be developed for each Enabling Objective in the Objectives Hierarchy. Whereas Criterion Tests measure the student's ability to accomplish the Terminal Objectives which represent the desired behavioral end products of instruction, Diagnostic Tests measure the accomplishment of the supporting skills and knowledges which contribute to the student's ability to perform the Criterion Objective. These are drawn from the Enabling Objectives (or sub-objectives) in the same way that the Criterion Tests were drawn from the Terminal Objectives. As with

the Criterion Tests, they should measure only the behavior which is to be taught.

Like the Criterion Objective tests, Diagnostic tests are developed for the Performance Evaluation Plan for use during Validation to identify problem areas and to adjust instruction for unanticipated factors such as differences between trainees. Once formal Validation is complete, diagnostic testing can be used to pinpoint training system problems. Since the Criterion Tests will provide the primary indication of instructional validity on an ongoing basis, diagnostic tests will be applied in a discretionary manner based on student performance.

2.3 Methodology Summary

- a) Organize the available programmatic and experiment information into specific data items and establish them in an Experiment Database.
- b) Using the information collected in the Experiment Database, derive Task Hierarchies representing all tasks necessary to operate and maintain the experiment.
- c) Add Conditions, Standards of Performance, and other attributes to each task in the hierarchy.
- d) Classify all tasks for training purposes.
- e) Derive Terminal Objectives from the tasks-to-be-trained, and establish them into an Objectives Hierarchy. Develop Enabling Objectives as necessary for each Terminal Objective.
- f) Develop Criterion and Diagnostic Tests for the Training Objectives.

3.0 INSTRUCTIONAL PLAN DEVELOPMENT METHODOLOGY

Instructional Plan Development refers to a set of processes which define how developed instructional requirements are met as well as how training effectivity is to be measured. In so doing, these processes produce media functional specifications, lesson specifications, test plans, and sequenced lesson plans.

The major inputs to Instructional Planning are the comprehensive hierarchies of behavioral objectives and related test items produced during Training Objective Development. These hierarchies and tests specify the Terminal Objectives and component Enabling Objectives, skills, and knowledge for every task to be trained. During Instructional Plan Development, these objectives are allocated to training media, analyzed for their functional requirements, organized into lessons, and sequenced according to an overall instructional strategy. Inputs which aid these processes include PTC training guidelines and policies, resource constraints, crew position requirements, and trainee individual and group characteristics. Outputs from this effort allow both simulator (hands-on) and academic media to be developed as well as the supporting instructional aids and materials. Figure 3-1 illustrates the general Instructional Plan Development process in terms of inputs and outputs.

Objectives and tests identify what is to be taught. Instructional Planning determines how it will be taught, and how to determine if the instruction is effective.

3.1 Instructional Methods and Media (Defining the Active Learning Environment)

Once the training objectives have been defined for an experiment, along with the underlying skills and knowledges required, instructional planning can begin by choosing the methods and media to be used in teaching them. While methods and media of instruction will be discussed separately here, they cannot be considered separately when specifying the active learning environment within which the students will acquire the desired skills and knowledges. It is the proper combination of methods and media which yield the most cost-effective training.

MAJOR INPUTS

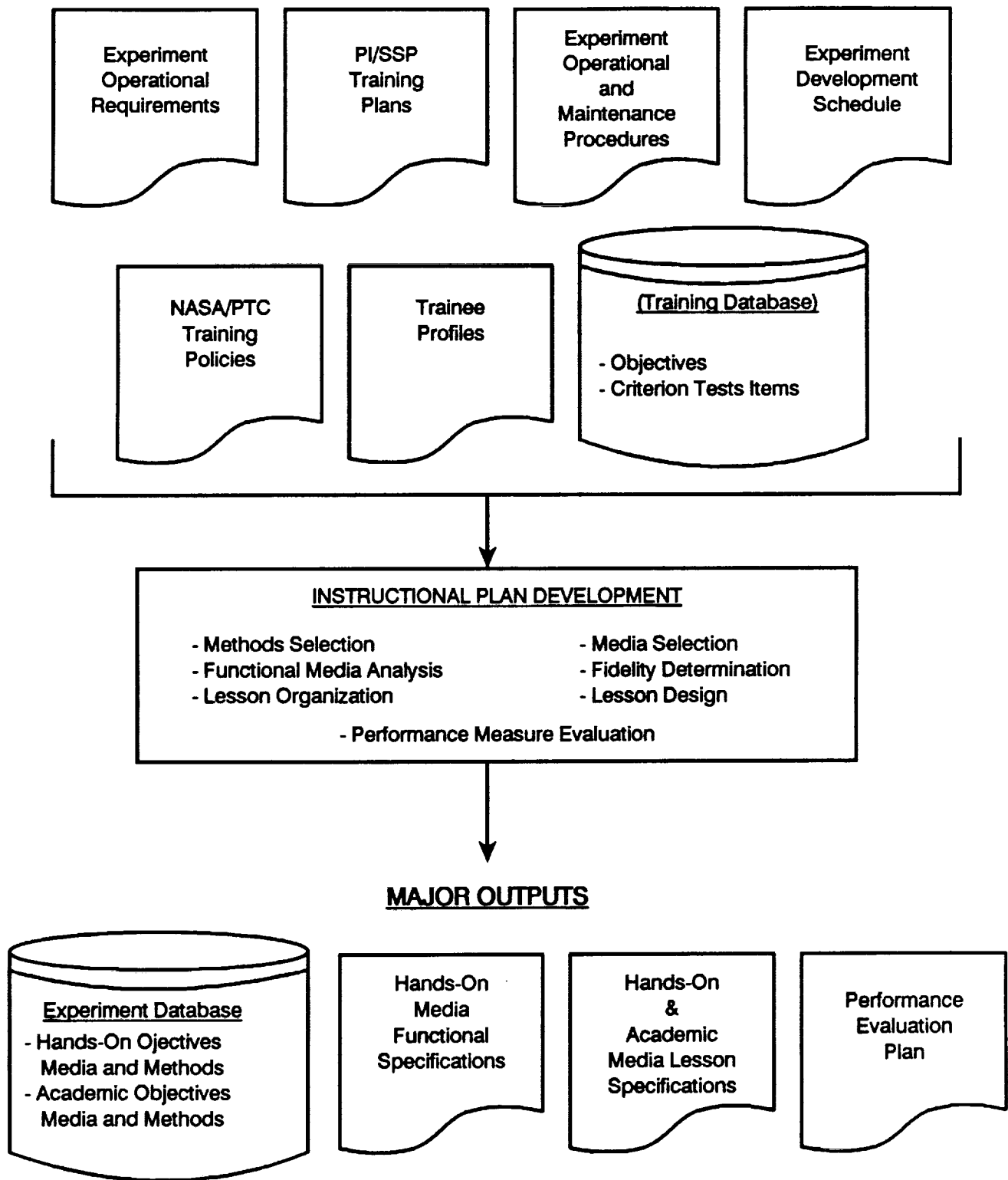


Figure 3-1. Instructional Plan Development (See Legend, Appendix C)

3.1.1 Instructional Methods and Strategies

IDENTIFY THE LEARNING TYPES ASSOCIATED WITH EACH OBJECTIVE

The most straight-forward way to determine optimal instructional methods for a given objective is to relate the behaviors involved to one or more types of learning. Examples of learning types include problem solving, rule using, and forming associations. A taxonomy or classification of learning types is shown in Table 3-1. While many learning taxonomies have been developed, this one has been edited to include the types of learning most applicable to payload training, in order of complexity. Since some instructional methods (and media) are more effective than others in aiding each type of learning, the types of learning involved in reaching an objective can help determine appropriate instructional methods for that objective. For example, a training Objective such as to "identify all instruments on a control panel" would involve a "forming association" type of learning. Possible instructional strategies to aid the student in making the proper associations would include presenting an exhibit, programmed questioning, or assigned reading. While there is no specific formula relating learning types to optimal instructional methods, a range of suitable candidate methods can be intuitively determined in this manner. To aid this intuitive technique, a matrix could be empirically derived over time from evaluations of actual training fielded at the PTC. This matrix would relate specific learning types with the range of instructional methods considered feasible for development and use at the PTC.

**DEFINE CANDIDATE INSTRUCTIONAL METHODS FOR EACH OBJECTIVE
BASED ON LEARNING TYPES**

Various instructional methods which have been selected as feasible alternatives for payload training are listed and defined in Table 3-2. Based on the learning types associated with each objective, a preliminary survey of these options should produce a range of candidate methods for each objective.

TYPE	PERFORMANCES RELATED TO DIFFERENT TYPES OF LEARNING
<p align="center">Forming Associations</p>	<p>Involves association, naming, or responding to a specific input (stimulus). The person associates the responses with a specific input only. The response may be vocal, subvocal (say-it-to-yourself), written, or motor. Examples: Naming objects you see; stopping at a red traffic light.</p>
<p align="center">Forming Chains</p>	<p>Involves recalling sequences of actions or procedures which must be recalled in a specific order. In a chain, the response to one input becomes the input to the next response. May involve chains of verbal responses or chains of motor responses. Examples: Verbal chain: reciting a memorized poem; stating a rule. Motor chain: tying a shoelace; starting an aircraft engine.</p>
<p align="center">Making Discriminations</p>	<p>Involves making different responses to the various members of a particular class; being able to distinguish among input information sources or types; and then to respond appropriately to each. Example: Recognizing the differences among similar gauges on an instrument panel and reacting appropriately with a vocal, subvocal, written, or motor response.</p>
<p align="center">Making Classifications</p>	<p>Involves responding in a single way to all members of a particular class of observable or abstract events. This involves recognizing the essential similarity among a class of objects, people, events or abstractions, and recognizing the differences which separate those objects, people, events, or abstractions which are not members of the class. Example: Classifying aircraft as being tactical, fighter, transport, etc.</p>
<p align="center">Using Rules</p>	<p>Involves applying rules to a given situation or condition by responding to a class of inputs with a class of actions. A rule states the particular relationship between two or more simpler concepts. It is helpful to think of rules as "if-then" statements. Example: If a metal rod is heated, then it will expand.</p>
<p align="center">Problem Solving</p>	<p>Involves comparing previously learned rules to create a higher order rule. Example: Troubleshooting a malfunction in an aircraft radar system. Many rules are involved in tracking down the specific malfunction.</p>
<p align="center">Performing Skilled Motor Acts</p>	<p>Involves manipulative tasks which require the smooth, integrated use of eyes and hands. Often this skill entails variation in the actions, where one action will be dependent on the results of other actions. Examples: Making a sensitive adjustment that requires precise timing; shooting a rifle accurately; driving a golf ball.</p>

Table 3-1. Classification of General Learning Types Applicable to Payload Training

	METHOD	DEFINITION
PRESENTATION METHODS	Lecture	A formal or semiformal oral presentation of information by a single individual on facts, concepts, problems, relationships, rules or principals presented orally either directly (as by classroom instructor) or indirectly (as by tape recordings, film, or TV).
	Demonstration	Presentation or portrayal of a sequence of events to show a procedure, technique, or operation; frequently combines an oral explanation with the operation or handling of systems equipment or material. May be presented directly (as by a classroom instructor) or indirectly (as by film, or TV, or by tape recording, if oral only).
	Exhibit	A visual or print display used to present information; for example, actual equipment, models, mockups, graphic materials, displays chalkboard, projects, images or sand table.
	Indirect Discourse	Verbal interaction among two or more individuals which is heard by the student; may be a dramatization, such as role playing, or a dialogue between panel members, or a teaching interview (a question and answer session between instructor and visiting expert).
	Assigned Reading	Printed verbal materials such as books, periodicals, manuals, or handouts. Readings may be course-assigned or self-assigned.
	Teaching Interview	Question and answer session between the instructor and a visiting expert following a highly structured plan.
STUDENT VERBAL INTERACTION METHODS	Questioning	A presenter-controlled interactive process used to emphasize a point, stimulate thinking, keep students alert, check understanding, or review materials. Questioning may be direct, as by a classroom teacher, or may be designed into a film or television presentation.
	Programmed Questioning	A presenter-controlled interactive process used to systematically demand sequence of appropriate student responses; may be used directly (as by a instructor in a classroom) or indirectly (as by programmed booklets or teaching machines, including computers).
	Student Query	The provision by which students are given the opportunity to search for information, as by questioning a classroom instructor, tutor, coach, or an appropriately programmed computer.
	Seminar	A peer-controlled group interactive process in which task- or objective-related information and experience are evoked from the students. Questions may be used to evoke student contributions, but the seminar is distinguished from questioning.
	Discussion	An instructor-controlled interactive process of sharing information and experiences related to achieving a training objective.
KNOWLEDGE APPLICATION METHOD	Performance	Student interactions with things, data, or persons, as is necessary to attain training objectives; includes all forms of simulation (for example, games and interaction with hardware simulators) and interaction with actual equipment or job materials (for example, forms). Performance may be supervised by classroom instructor, tutor, coach, or peers to provide needed feedback.
	Case Study	A carefully designed description of a problem situation, written specifically to provoke systematic analysis and discussion.

Table 3-2. Definition and Classification of Instructional Methods Applicable to Payload Training

SCREEN CANDIDATE METHODS ON THE BASIS OF STUDENT TRAITS,
COST, RESOURCES, AND OTHER EXTERNAL FACTORS

From the initial group of candidate instructional methods, a further selection may be made by consideration of factors such as student individual and group characteristics, cost, and resources: Student Characteristics - Overall, the preliminary flight and ground crew profiles (high aptitude, high motivation, see Appendix A, Flight and Ground Crew Characteristics) imply a curriculum which is learner-directed and learner-paced. Applicants with higher mental aptitude and the capability for independent field work may be expected to take an active role in their learning, supply much of their own motivation and require less positive reinforcement. These considerations, as well as the need to accommodate individual trainee differences, recommend that the instructional methods chosen must be flexible enough to permit individual students to proceed at different rates through a training sequence and/or to repeat segments until they are mastered. This requirement may eliminate certain methods from consideration.

Cost and Resources - For the PTC, the availability of instructors, facilities, equipment and materials in reference to time allotted for instruction, student load, and class size are factors which will affect the cost of instruction and therefore, selection of an instructional method. While the primary instructional criterion should be training effectiveness, selection between methods of equal value should be on the basis of cost.

Figure 3-2 illustrates the procedure for selection of instructional methods. The output of the methods selection process will be a set of learning types and candidate instructional methods stored as attributes of each objective, in the Experiment Database.

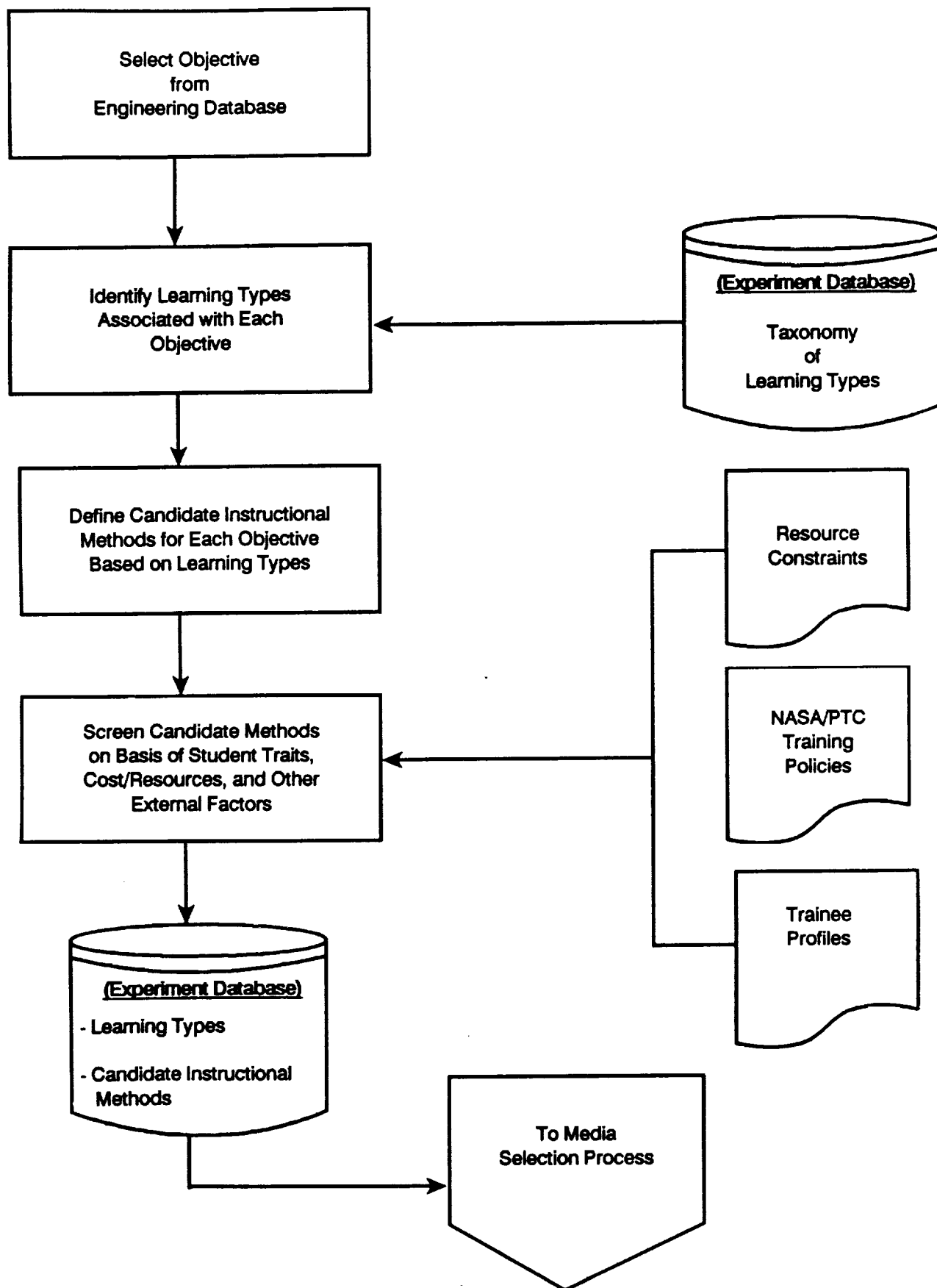


Figure 3-2. Selection of Instructional Methods

3.1.2 Instructional Media Selection

Besides instructional strategies, the most important aspect of the active learning environment is the medium, or means through which the student will be given information. These means can range from classroom lecture, to a workbook, to simulators or training on actual system (payload) equipment. Appropriate media must be selected for each objective based primarily on training effectiveness for that objective. Evaluation of the effectiveness of a medium must consider the ways in which it will accommodate

- a) Presentation of the information
- b) Use or practice
- c) Feedback to the student.

If alternative media have equal effectivity in each of the preceding areas, then the choice between them should be made on the basis of cost, availability, maintainability, or other external factors.

Candidate media are established for each objective by relating characteristics of the instructional requirements which they represent to attributes of the media alternatives. However this relational process is performed, it is a prime candidate for proceduralization and automation. A number of automated models have been proposed and developed, such as the Automated Instructional Media Selection (AIMS) system developed by Kribs, Simpson, and Mark (1983). The AIMS system is designed to relate up to 90 instructional characteristics, such as strategy, crew interaction, or degree of feedback, to up to 90 instructional media. The methodology presented in this study is a manual one; it is given primarily for illustrative purposes. The most important factor in media selection, however, whatever the methodology, is that it be based on instructional requirements and training needs.

IDENTIFY HANDS-ON VERSUS ACADEMIC MEDIA OBJECTIVES

The first step in media selection is to review the training objectives hierarchies in order to identify those objectives that will require hands-on training or practice. This shall be accomplished by examining the behavioral statement and conditions for each objective. All objectives requiring real or simulated operational equipment shall be designated as hands-on objectives (for example, objectives requiring visual, auditory, motion,

environmental, and other cues that must be presented in some manner of computer-driven live enactment), and the remainder designated as academic objectives, requiring academic type instruction. This classification will be recorded as an objective attribute in the Experiment Database.

This discrimination will later serve to channel the objectives to two development flows. Both academic and hands-on objectives will go to Syllabi Development for use in constructing Lesson Specifications, and training support materials such as training scripts. Hands-on objectives will also be used to develop simulator functional requirements and specifications.

ANALYZE OBJECTIVES AND OTHER DETERMINANTS TO ESTABLISH
CANDIDATE ACADEMIC AND HANDS-ON MEDIA

The hands-on objectives will be analyzed by relating the instructional requirements which they represent to attributes of hands-on media. Likewise, the academic objectives will be analyzed by comparison with academic media characteristics. The result of this activity will be a list of candidate media for each hands-on or academic objective stored with each objective in the Experiment Database. Table 3-3 lists some representative examples of both kinds of media.

CAPABILITY CLASSIFICATION

1. Full Scale Mockup
2. Static Procedures Trainer
3. Computer-Driven Trainers
 - Part Task
 - Whole Task
 - "Billboard" Type
4. Actual Experiment Equipment
5. Hybrid (Actual/Simulated Equipment)

ACADEMIC MEDIA

1. Computer Based Training
2. Videotape
3. Workbook
4. Scale Model
5. Lesson Guide
6. Slide Show

Table 3-3. Hands-On Versus Academic Media Classification

Table 3-4 represents the large number of potentially available media divided into five major groups. All of the specific media examples given are considered possibilities for payload training. They do not represent a complete list but demonstrate the range of possibilities, so that the factors which make a particular medium effective will be more apparent. The following factors should be considered in order to determine suitable media for each learning objective:

Compatibility with Types of Learning - Simply put, most types of learning are more effectively taught with some media rather than others. It is possible to identify suitable media, or eliminate them from consideration, on the basis of the types of learning associated with a particular objective. In most cases, this is clear, such as the greater effectivity of practicing motor skills with an individual tutor, rather than with an instructor in a lecture hall. It is, however, a good method for establishing as broad a range of candidate media as possible, before elimination through other means. Table 3-4 is provided as an aid to this process. It relates the types of learning listed in Table 3-1 to representative instructional media. By comparing the learning types of the objective under consideration to the Table, inappropriate options can be ruled out.

LEARNING OBJECTIVES AND ELEMENTS OF LEARNING STRATEGY	Associations & Discriminations			Verbal Chains			Motor Chains			Classifying, Rule Using & Problem Solving			Perceptual Motor Skills			Attitudes, Opinions & Motivations			
	Presentation	Practice	Feedback	Presentation	Practice	Feedback	Presentation	Practice	Feedback	Presentation	Practice	Feedback	Presentation	Practice	Feedback	Presentation	Practice	Feedback	
Representative Instructional Media																			
CLASSROOM INSTRUCTOR																			
Lecturer	YES	NO	NO	YES	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	P
Demonstrator	YES	NO	NO	YES	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	P
Tutor/Coach	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	P
Teaching Interview	YES	NO	NO	YES	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	P
INSTRUCTIONAL AIDS																			
Overhead Projector	YES	P	P	YES	P	P	YES	P	P	YES	P	P	YES	P	P	NO	NO	YES	P
35mm Slides	YES	P	P	YES	P	P	YES	P	P	YES	P	P	YES	P	P	NO	NO	YES	P
Chalkboard	YES	NO	NO	YES	NO	NO	YES	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
8mm Movies (silent loop)	YES	NO	NO	YES	NO	NO	YES	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
MULTIMODAL MEDIA																			
Prenarrated Slides (with stop)	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	P
Prenarrated Filmstrip	YES	P	P	YES	P	P	YES	P	P	YES	P	P	YES	P	P	YES	P	YES	P
Slider/Workbook/Audio Cassette	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	P
Movies (sound)	YES	P	P	YES	P	P	YES	P	P	YES	P	P	YES	P	P	YES	P	YES	P
TV (cassette)	YES	P	P	YES	P	P	YES	P	P	YES	P	P	YES	P	P	YES	P	YES	P
PRINT																			
Books	YES	P	P	YES	P	P	YES	P	P	YES	P	P	YES	P	P	NO	NO	YES	P
Computer (words & numbers only)	YES	P	P	YES	P	P	YES	P	P	YES	P	P	YES	P	P	NO	NO	YES	P
Programmed Instruction Booklet	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	P
Microfiche	YES	P	P	YES	P	P	YES	P	P	YES	P	P	YES	P	P	NO	NO	YES	P
PEER																			
Role Playing	P	P	P	NO	NO	NO	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	YES
Discussion Group	P	P	P	NO	NO	NO	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	YES
Tutor/Coach	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES
SIMULATION																			
Actual Equipment Trainer	NO	P	P	NO	NO	NO	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES
Gaming	NO	P	P	NO	NO	NO	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	YES
Interactive Computer (plasma terminal)	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES
Case Study	NO	P	P	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	YES
ACTUAL END ITEMS	NO	P	P	NO	NO	NO	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES

CODE:
P - Under certain circumstances, effective use is possible,
or there are inherent limitations in the effective use
YES - is effective
NO - is not effective
N/A - use of the media is inappropriate

Table 3-4. Applicability of Instructional Media of Learning Types

Top Level Fidelity Requirements - One of the primary determinants of training media is the degree to which the training environment must resemble that of the job. If it is determined for example, that procedures training for an experiment will require a close correspondence to the spatial relationships of the actual system, then certain types of media such as classroom or CBT will automatically be ruled out. Generally, tasks that have concrete inputs requiring concrete responses, such as sensor calibration, would need a learning environment that more closely resembles that of the task. Activities with abstract informational inputs and outputs such as learning to compute resource utilization schedules would be less likely to need a high fidelity learning environment.

Broadband fidelity requirements can be used as an aid to determining suitable training media. This should not be confused, however, with the more detailed process of determining the fidelity requirements for the selected media. This will be treated separately, as a further step in the media selection process.

Real World Constraints - Other reasons to select or reject candidate teaching media include the real world conditions under which training will be developed and conducted. These include the:

Target Population and their probable range of aptitude, experience, skills, and knowledge. If for example, a student group was known to possess limited reading and writing skills, then one possible response might be to include training to bolster reading and writing skills (remedial training). Another approach might be to limit the use of text as an instructional medium and rely more on graphics to transmit ideas (compensatory training). In the case of payload operations, given the anticipated characteristics of both the flight and ground crews (see Appendix A, Flight and Ground Crew Characteristics) these considerations will probably not find much applicability; though the students' initial experience, skills and knowledges will certainly affect instructional methods and content.

Availability of Time to develop and start instruction may influence media selection and should be considered. A training medium for example, which is not currently in use or planned for use at the PTC may require more time to develop than schedules permit. Likewise, instruction for behind-schedule experiments may have to be developed within a time frame which will not allow use of certain media (such as classroom instruction might have to substitute for CBT courseware).

Resources such as instructors, equipment, or facilities may preclude or encourage selection of media. This is almost always a

real world consideration. It is likely that the PTC will develop and use certain representative media from each of the major groups in Table 3-5 in order to accommodate the most likely or common methods of instruction. Therefore the training developer will have programmatic guidelines as to the media of choice for a given instructional method. Slide presentations for example, could be a favored option over videos. Likewise, resource limitations may preclude for example, the use of individual tutors over other, less personnel-intensive media.

Student Load and its relation to available resources can influence media selection, in that the capacity to develop or use certain media or instructional materials may be overloaded in one area and under-utilized in another. Also, certain media may not be cost-efficient for use by limited numbers of students requiring instruction. CBT courseware for instance, may not be a good choice to present specialized or one-time material of interest to only a handful of trainees.

Cost of instruction definitely varies from one medium to another. This includes the cost of procuring or developing the media, associated courseware and instructional materials, as well as costs for operation and maintenance. While the costs of certain media or features of media may automatically preclude their use, in many cases a tradeoff will have to be made between somewhat lower cost on one hand, versus somewhat higher training effectivity on the other. All things being equal (such as training effectiveness) the least expensive media should be chosen. These and other real-world constraints will be revisited in Simulator Requirements Derivation, when an overall simulator approach is determined for each experiment.

INSTRUCTIONAL MEDIUM GROUP	REPRESENTATIVE EXAMPLES
Classroom instructor with instructional aids - Classroom instructor - Instructional aids	Lecturer Demonstrator Tutor/Coach Overhead projector Film strip (silent) Film slides Chalkboard
Multimodal media	Prenarrated slides Prenarrated filmstrips Slide/workbook/tape recorder combinations Videotape
Print	Books Computer (words and numbers only) Programmed instruction booklets
Peer (or peer group)	Role playing Discussion groups Tutoring/coaching
Training devices and simulators	Computer Based Training (CBT) Actual equipment trainers Interactive computer (simulation) Training simulators

TABLE 3-5. Representative Range of Instructional Media Suitable for Payload Training

SCREEN CANDIDATE MEDIA BY COMPARISON WITH RECOMMENDED
INSTRUCTIONAL METHODS

The last step in media selection is to screen the candidate media by comparison with the candidate instructional methods listed in the Experiment Database. The training media and instructional strategies chosen must complement one another. In other words, the media must be capable of implementing the methods assigned to each objective as well as the techniques required by the lesson specifications. As an example, an objective such as to "disassemble the IR sensor to its component parts and check for corrosion" might have "demonstration" or "performance" designated as candidate methods. Suitable media with which to present the required training information could include a sensor mockup or exhibit. Media such as classroom or CBT, on the other hand, may not provide the requisite functionality, depending on the stage of training and other factors. If the objective is to "identify and name all parts of the IR sensor assembly," however, suitable instructional methods could be "lecture" or "student query," in which case the classroom or CBT environment would be adequate.

The output of this activity will be a set of recommended media and methods for each objective, linked to the appropriate objective in the Experiment Database, and including rationales for all media selections made. The hands-on media selection will be further examined to determine the required functionalities needed to train for their respective objectives. These collective functional requirements will be used to develop hands-on functional specifications for each type of selected hands-on media.

For the purposes of the envisioned payload training, the various types of hands-on media are distinguished primarily by their functional specification. Consider for example, the hands-on media types listed in Table 3-3. Besides the billboard trainer (which is not particularly applicable to payload training) the other choices are distinguished from each other primarily by their fidelity and functionality. It is therefore possible to allocate an objective simply to hands-on media, and allow the associated fidelity and functional requirements to complete the media definition. The final hands-on media functional specification will in fact group a set of training objectives with compatible functional and fidelity requirements together to fully define a media for learning. This media can then be designated as a procedures trainer, mockup, or whatever other label applies.

By aligning each experiment simulator function along functional and fidelity rather than PTC architectural lines (Part Task, Module, Consolidated, etc.) we separate the issue of media classification from concerns with scheduling, resource allocation, etc. Individual experiment simulators can be developed to a certain level of fidelity, then housed in whatever trainer is most convenient for that increment (subject to the trainer's designated instructional role, and interface and support capabilities). The placement of individual experiment simulators within the PTC for resource scheduling or other reasons, will be taken up in the Simulator Requirements Derivation activity when a simulator approach for each simulator is addressed.

Academic media selections will be re-examined during syllabi development by analysis of their common characteristics when grouped into lessons. Final academic media selections and functional academic media specifications will be made at that time. Figure 3-3 illustrates the procedures for Instructional Media Selection.

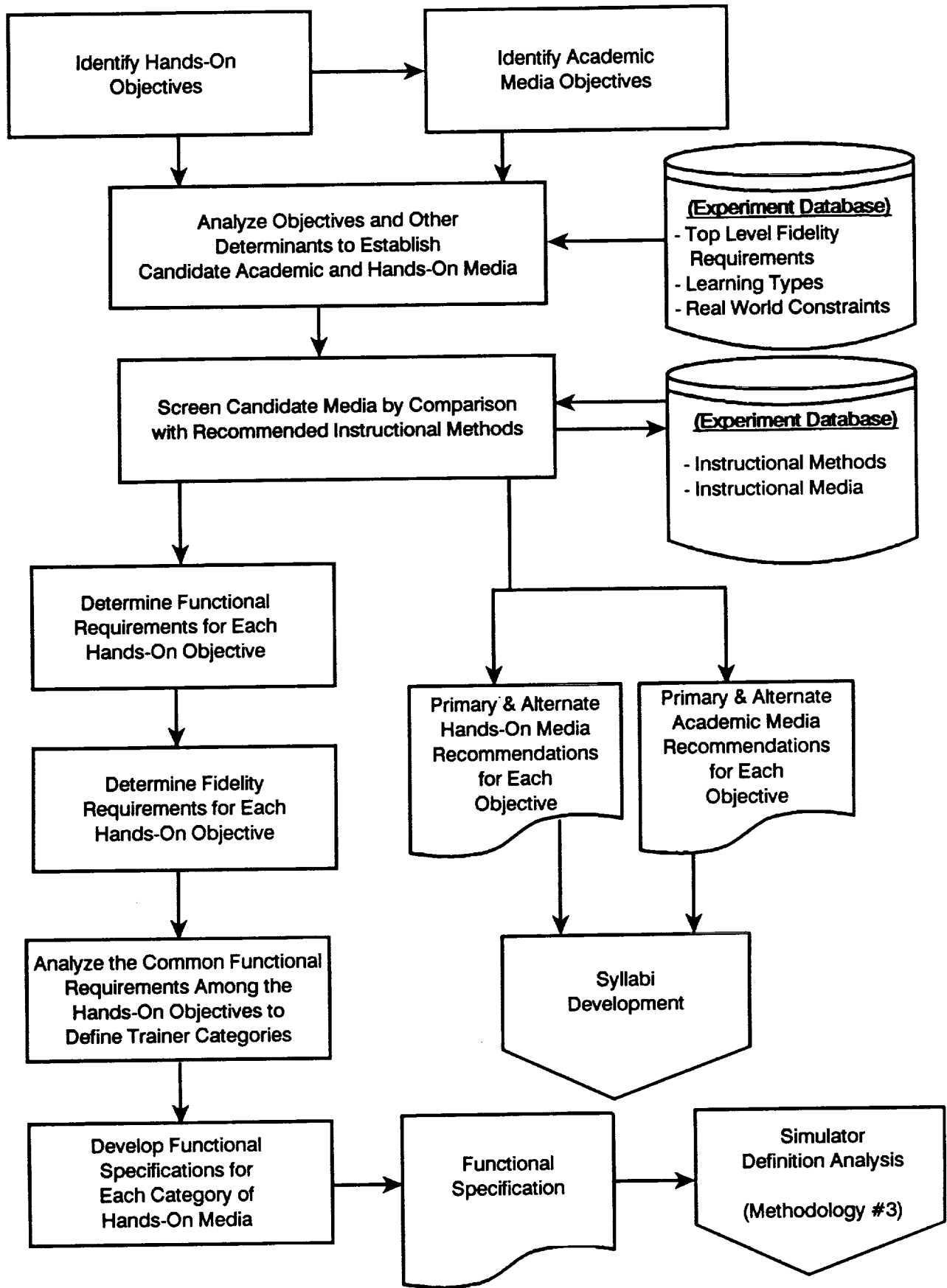


Figure 3-3. Instructional Media Selection

3.1.3 Instructional Methods and Media Procedures Summary

METHODS

- a) Identify the learning types associated with each Training Objective.
- b) Define candidate instructional methods for each Training Objective, based on learning type.
- c) Screen candidate methods based on student profiles.

MEDIA

- a) Identify hands-on versus academic media Training Objectives.
- b) Define candidate media for each objective based on:
 - Compatibility with learning types
 - General fidelity requirements
 - Student profiles
 - Time constraints, student load, cost
 - Development or training resources
 - Compatibility with instructional methods.

3.2 Hands-On Media Functional Requirements and Functional Specifications

DETERMINE SIMULATOR FUNCTIONAL REQUIREMENTS FOR EACH HANDS-ON MEDIA OBJECTIVE

As a preliminary step towards establishing media functional specifications, each objective will be analyzed separately to determine the functional requirements which the training media must satisfy in order to meet the training requirements. These functional requirements will be used later to establish the functional specifications for each media type employed in training. Inputs to functional requirements include Task Analysis data (previously developed), as well as Lesson Specifications which will be generated as part of Syllabi Development.

Figure 3-4 illustrates the flow of Functional Requirements and Specifications Activities.

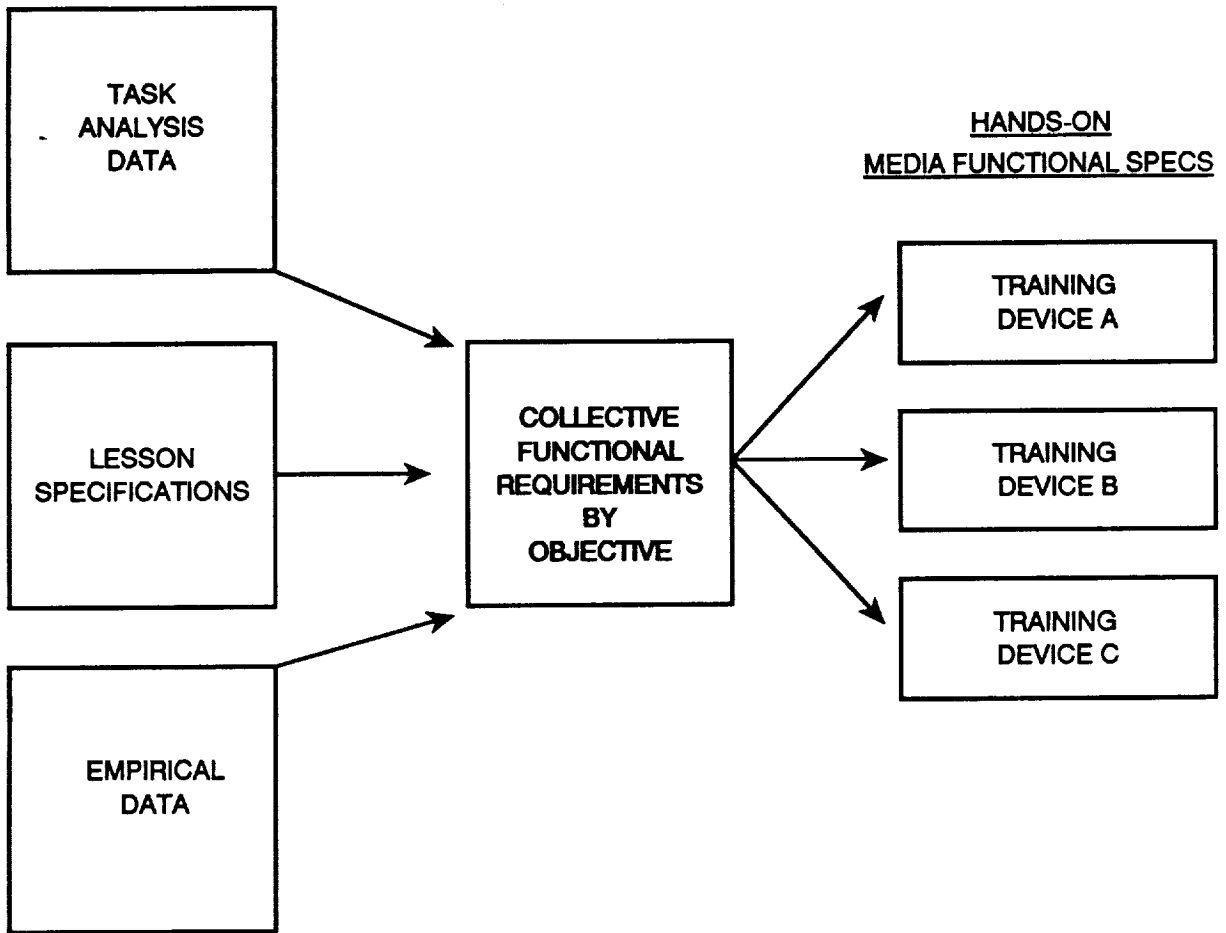


Figure 3-4. Functional Specifications and Requirements Derivation and Allocation to Media Types

A third input, very important to the development of training device requirements is empirical data on the ways in which factors both extrinsic and intrinsic to the training task interact with the device characteristics needed for cost-effective training. These factors include task difficulty, trainee sophistication, task type etc. (see Table 3-6). Empirical data on these relationships as they specifically relate to payload training are scarce. While the functional requirements derivation process explained below should provide a reasonable first cut at how to effectively train for specific tasks, systematic efforts to relate training effectiveness to specific instructional strategies and device features will be necessary if the methodology is to evolve and achieve optimal results in the payload training application.

<p>1. Task Type</p> <ul style="list-style-type: none"> - Operations - Maintenance - Others 	<p>5. Stage of Training</p> <ul style="list-style-type: none"> - Introduction - Procedural Training - Familiarization Training - Skill Training - Transition Training
<p>2. Task Difficulty</p>	
<p>3. Specific Skills Required by Task</p> <ul style="list-style-type: none"> - Motor - Perceptual - Cognitive - Others 	<p>6. User Acceptance</p> <ul style="list-style-type: none"> - Instructors - Students
<p>4. Trainee Sophistication</p> <ul style="list-style-type: none"> - Novice - Intermediate - Expert 	<p>7. Use of Instructional Features</p>

Table 3-6. Variables Which Interact With Fidelity

3.2.1 Hands-On Media Functional Requirements

Fidelity is a two-dimensional measurement of similarity between the training and operational settings in terms of the physical (how a device feature looks) and functional (how a device feature works) characteristics of the simulated system. Functional fidelity requirements are driven by training requirements which specify what must be learned. In order to satisfy the training requirements, the trainer must be capable of presenting the cues necessary to elicit or prompt the desired behavior. Determination of appropriate physical and functional fidelity levels must be based on a determination of the cues and features which will best teach the desired skills and learner strategies. Though the two aspects of fidelity are extremely interactive, functional fidelity requirements (representing training requirements) should guide physical fidelity requirements, so that the physical fidelity (and cost) of the trainer may be the minimum sufficient for effective learning.

The goal of fidelity analysis is to determine the most cost-effective degree of correspondence between the learning environment and that of the task. At first, this involves simply choosing representative features of the experiments' displays and controls for inclusion in candidate training media, such as a "billboard" type simulator, mockups, single experiment simulators, etc. This selection process should be based on an analysis and understanding of the specific cues and features of the experiment necessary to successfully perform each task, guided by the need to satisfy overall training objectives.

Task Analysis Data:

USE TASK ANALYSIS DATA TO HELP DETERMINE THE OPERATIONAL CUES AND FEATURES REQUIRED TO ACCOMPLISH EACH OBJECTIVE

The predominant functional requirements for each Training Objective will be determined from examination of the data collected during task analysis. Analysis of this information should indicate what kind and how accurate the sensory information presented to the student must be to accomplish the required training.

It is both possible and useful to refer to this determination of a training device's specifications as a fidelity analysis, since the fidelity of a training device is defined by its physical and functional specification. When all the elements of the operational environment have been analyzed to determine their

minimum required fidelity in the training environment, the result is a training device specification which will include only those elements of the actual job required for training. The training device built from such a specification will be more cost-effective than the same device built according to the "shotgun" approach which strives to blindly duplicate all operational equipment features.

Task Analyses are performed to provide information to many aspects of the instructional development process. This data is used in media selection, instructional strategy development, trainee selection, and many other areas besides fidelity requirements decision making for training simulators. In order to focus on the information required for fidelity decisions, a subset of the total task analysis information may be reorganized into a format which will allow easy access to relevant information. A sample format is shown in Figure 3-5. This reorganization of data can easily be automated for the training analyst by a database utility. Within each objective, the utility would display the information separately for each appropriate task and subtask in the correct order of execution.

Identification/Title	
TASK:	
PARENT OBJECTIVE:	
SUBTASKS:	
Task Description (In Operational Environment)	
ACTIONS REQUIRED:	
EQUIPMENT REQUIRED FOR TASK: (Including drawings or references to drawings)	
Controls	
Displays	
Tool/References	
Internal Components (For test, repair, or manipulation of experiment hardware)	
CUES:	
Display Information, Format, Resolution	
Auditory	
Other	
SKILLS/KNOWLEDGES (Prerequisite skills/knowledges as well as those to be taught)	
Physical	
Perceptual/Motor	
Cognitive	
STANDARD OF PERFORMANCE:	
CONDITIONS:	
Initiating Conditions	
Terminating Conditions	
External Constraints	
Relevant Contingencies	
Malfunctions	
CFD RATINGS:	Rationale
Criticality	
Frequency	
Difficulty	

Figure 3-5. Sample Format for Task Analysis Data Used in Fidelity Determinations

For fidelity determinations, a decision will be made as to how each component of the operational equipment will be represented in the simulator. Therefore to determine required fidelities, the fidelity task database must contain a complete description of each task including the displays and controls required (or references to drawings, lists, etc.), the detailed actions required, Conditions and Standards of Performance, relationships with other task elements, as well as underlying skills and knowledge. All displays and controls should be indicated on line drawings or photographs to determine if the layout of controls is an important factor for a particular task (this kind of information will be important when specifications are assembled for each type of media). Some components will not have to be represented in the simulator because they are not involved in training tasks. On the other hand, some controls and displays must be represented because they provide locational cues to the ones which will be trained.

If the preliminary task analysis was thorough, the training analyst should be able to use that data to determine the cues and experiment features that are utilized in the operational environment to perform each task. An approximate level of required fidelity can then be defined in terms of the training device features and capabilities required to provide the same cues and features in a training situation. In order to refine this rough estimate, CFD (Task Criticality, Frequency, Difficulty) ratings for each task can be used to gauge the precision with which the required cues and features will be replicated. This refinement process will be described in the Information Processing Demands Section.

Task Analysis data, reformatted for the purpose of Fidelity Analysis (Figure 3-5) should contain enough information to enable determination of the operational cues and experiment features necessary to perform, or learn to perform, each task. Below are listed a few key points to consider when deriving a set of trainer functional requirements from the task data:

a) Displays: What information relative to each task does the display provide? What information resolution is demanded by the task? What display characteristics and formats are used?

b) Non-Display Inputs: These include auditory, textual, or other information modes which convey information to the experiment operator. What task-relevant information do they provide? What resolution is demanded by the task?

c) Controls: Experiment controls may or may not provide tactile and other cues. Furthermore, such cues may or may not be critical for skill acquisition. Characteristics such as sensitivity, resolution, and feel forces must be evaluated in terms of

providing the feedback required for the learning and performing of specific tasks. Stimulus - Response interactions between the controls and displays help determine the required fidelities of both.

d) Layout: Is the control and display configuration used as a cue to help locate certain devices?

e) Actions Required:

- Perceptual and Motor Skills: If a task demands the use of an "input and output" type skill, the experiment simulator must allow that skill to be exercised. If perceptual, the simulator must provide sufficient perceptual cues to allow the skill to be demonstrated. Likewise for motor skills, the simulator functionality must be enough like the actual experiment to allow skill acquisition.

- Physical Proficiencies: As above, the simulator functionality must be such as to allow the development or retention of physical skills, if appropriate to the stage of training and the overall training plan.

- Performance Criteria: The standard by which trainee performance is judged is often a good indicator of the accuracy required of the learning cues. For example, if a trainee is required to adjust a sensor to within .05 degrees of arc, the resolution of the feedback cue, as well as the simulation generating the cue, must be sufficient to allow this level of accuracy. In addition, the means by which performance is to be measured must be enabled by the trainer functionality.

f) Internal Components and Layout: In addition to external appearances, what internal fidelity or capability requirements are levied by maintenance tasks, malfunctions, etc.?

g) Cognitive Skills Required: As above for physical, perceptual, or motor skills, the trainer must provide the necessary information to allow the performance of a cognitive task, as well as the means to express the behavior resulting from cognition.

h) Stage of Training: In general, the greater the correspondence of the learning environment with that of the task, the greater will be the transfer of learning. An important exception to this is during the early learning stages, when only a subset of the total job tasks have been trained. In this situation, the inclusion of environmental details extraneous to the task at hand can be distracting and confusing to the novice. A complex instrument panel, for example, with full functionality when used

to train the significance of only a few gauges may be confusing to the beginner.

In this case, it might be better to initially use a training device with only those gauges active which are needed for the training task. The others could be blocked from view, inactive, or simply not installed, depending on the total training for which the device is intended.

In the case of payload training, it may quite possibly prove to be more economical to produce one panel design for all training purposes than a unique design for each kind of trainer or training. If so, the temporary obstruction of superfluous panel assemblies would be an obvious alternative. Care must be taken, however, not to impair their utility as locational references. This may be ensured by placing simplified representations of the operational equipment (wallpaper) in front of the original panels.

Another example of the use of selective fidelity to accommodate trainee skill level would be when training crew members to properly interpret sensor imagery. During the early learning stages, the simulated image could be unrealistically simplified to enable easy identification of target phenomenon. Later, as student expertise increases, the images could be gradually enriched with "extraneous" details likely to be perceived by the actual sensor until the student can make the necessary discriminations under real world conditions.

i) Conditions of Performance: What malfunctions must be simulated? What are the malfunction symptoms? Are there operational contingencies such as resource sharing involved? What other activities must be simultaneously occurring? Are there peripheral cues indirectly associated with the task and the experiment which must be provided? What are the initiating and terminating conditions for the task?

j) CFD (Task Criticality, Frequency, Difficulty) Ratings: These ratings provide valuable information on the level of fidelity required for certain tasks. These ratings will be used to refine and crosscheck the initial fidelity determinations.

In general, when assembling functional requirements for an objective, the best approach is to start from zero. That is, rather than proposing an experiment replica, then subtracting unneeded features and capabilities; start with absolutely nothing, adding features and cues only as demanded by training requirements. For each task, a set of functional requirements can be assembled, based on an analysis of task data. Figure 3-6 depicts a sample format for organization of the functional

requirements. The purpose of this information will be to allow functional specifications to be written for each media type.

Identification/Title	
OBJECTIVE:	
Hardware/Data Requirements to Train Objective	
Equipment Required for Objective: (Including drawings or references to drawings)	
CONTROLS	
Required Appearance	
Required Functionality	
Required Durability (In terms of projected usage)	
Tactile/Feel Characteristics	
DISPLAYS	
Required Content, Level of Detail	
Required Format, Resolution	
Required Functionality, Response to Trainee Actions	
INTERNAL COMPONENTS (Test/Repair/Manipulation of Experiment Hardware)	
Required Appearance	
Required Functionality	
Required Durability (In terms of projected usage)	
Tactile/Feel Characteristics	
TOOLS/REFERENCES	
Required Appearance	
Required Functionality	
Required Durability (In terms of projected usage)	
Tactile/Feel Characteristics	
AURAL/OTHER CUES	
General Characteristics	
INSTRUCTIONAL FEATURES	
Augmented Feedback	
Feedback to Instructor	
Performance Evaluation Capabilities	
Live Instructor	
Detailed Rationale for Fidelity/Functional Requirements Listed Above:	

Figure 3-6. Sample Format for Simulator Functional Requirement Form

Lesson Specifications:

MODIFY THE DERIVED OPERATIONAL CUES AND FEATURES IF NECESSARY, WITH INPUTS FROM APPROPRIATE LESSON SPECIFICATIONS

Lesson Specifications will provide another input for the determination of media functionality. If for example, a full physical and functional simulator was chosen as the medium to train certain objectives, the lesson specification covering those objectives might specify certain malfunctions or other abnormal behaviors to which the trainee would be introduced during the training scenario. The capability to provide these cues would comprise a functional requirement to be included in the hands-on media Functional Specification. Other possible inputs include instructional media features, such as specific data feedbacks to the training instructor during the performance of a training scenario. These instructional features would also contribute to the Functional Specification.

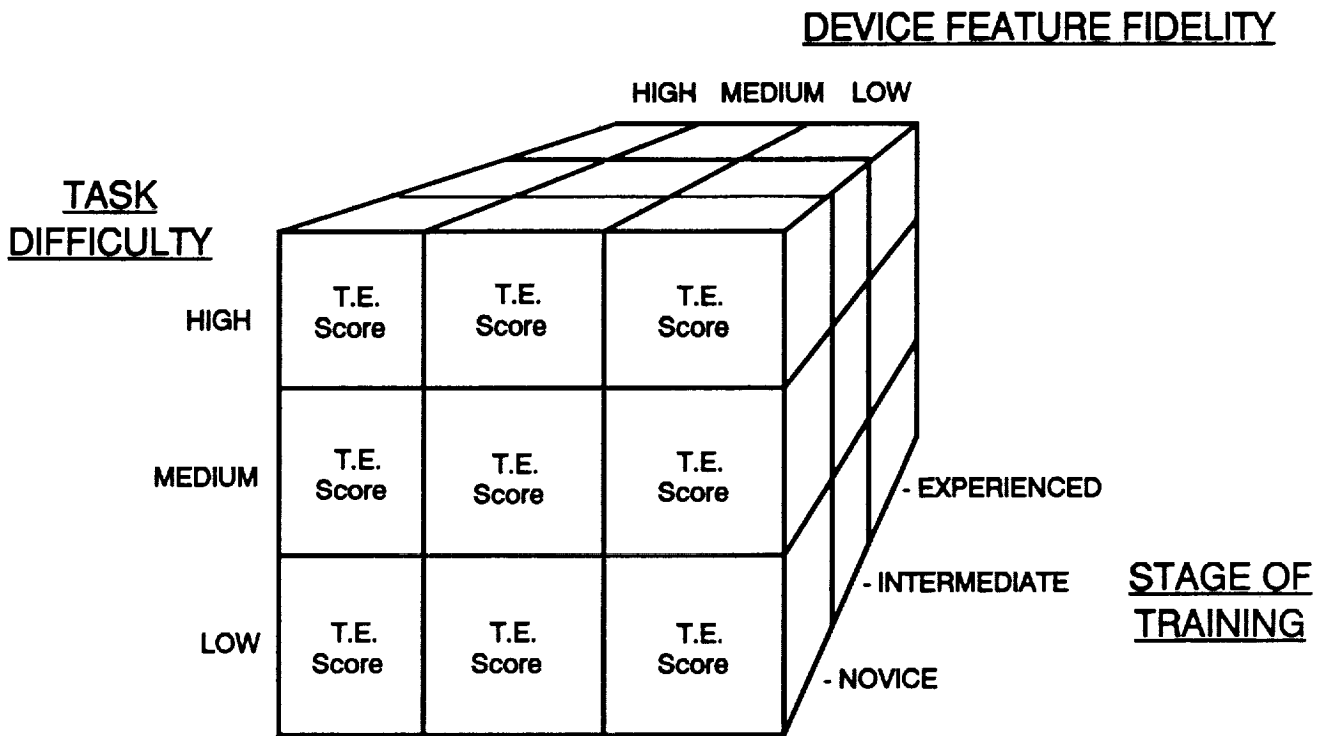
Empirical Studies:

USE EMPIRICAL STUDY RESULTS (IF AVAILABLE) AS AN INPUT FOR MINIMUM REQUIRED LEVELS OF PHYSICAL AND FUNCTIONAL FIDELITY

Empirical data is another input useful for determining the optimal required fidelity for the various components of a training simulator. The goal is to be able to determine the minimum fidelity levels necessary for cost-effective training under a variety of circumstances. This is a subset of the overall task of developing empirical data on the effects of all internal and external variables on training effectiveness. Table 3-6 lists many training variables which can have an effect on the level of fidelity required for cost-effective training. Studies should be devised to generate empirical data which will delineate the effect on required fidelity levels for each variable. These studies will be an ongoing activity, consisting primarily of an analysis of student performance data collected during training,

followed by application of the analytic results to the development process in order to iteratively refine the development methodology. Rather than an independent research effort, these studies should be integrated with the normal flow of training at the PTC.

The studies must isolate as much as possible, the effects of each variable by comparison of training scenarios which are as alike as possible except for the variable of interest. Combinations of different fidelity-interactive scenario variables should also be tested to determine how their effects are modified by each other. As an example, Figure 3-7 illustrates an organized test matrix which could be used to determine the influence of task difficulty on required fidelity levels at various stages of training. Each block represents a performance evaluation of training scenarios with different combinations of fidelity levels, difficulty, and trainee experience. By evaluating the training effectiveness evinced by each scenario, interactions between fidelity level and other training variables can be studied to determine optimum fidelity levels under various circumstances. Results from systematic studies such as these would be compiled into an iterative simulator fidelity database which could then provide empirically-based answers to fidelity questions. As understanding (and the database) grows, more accurate predictions of required fidelity levels will be used to reap maximum efficiency from training devices.



Note: TE = Training Effectiveness (1-10)

Figure 3-7. Sample Matrix for a Fidelity Study of Task Difficulty at Various Training Levels

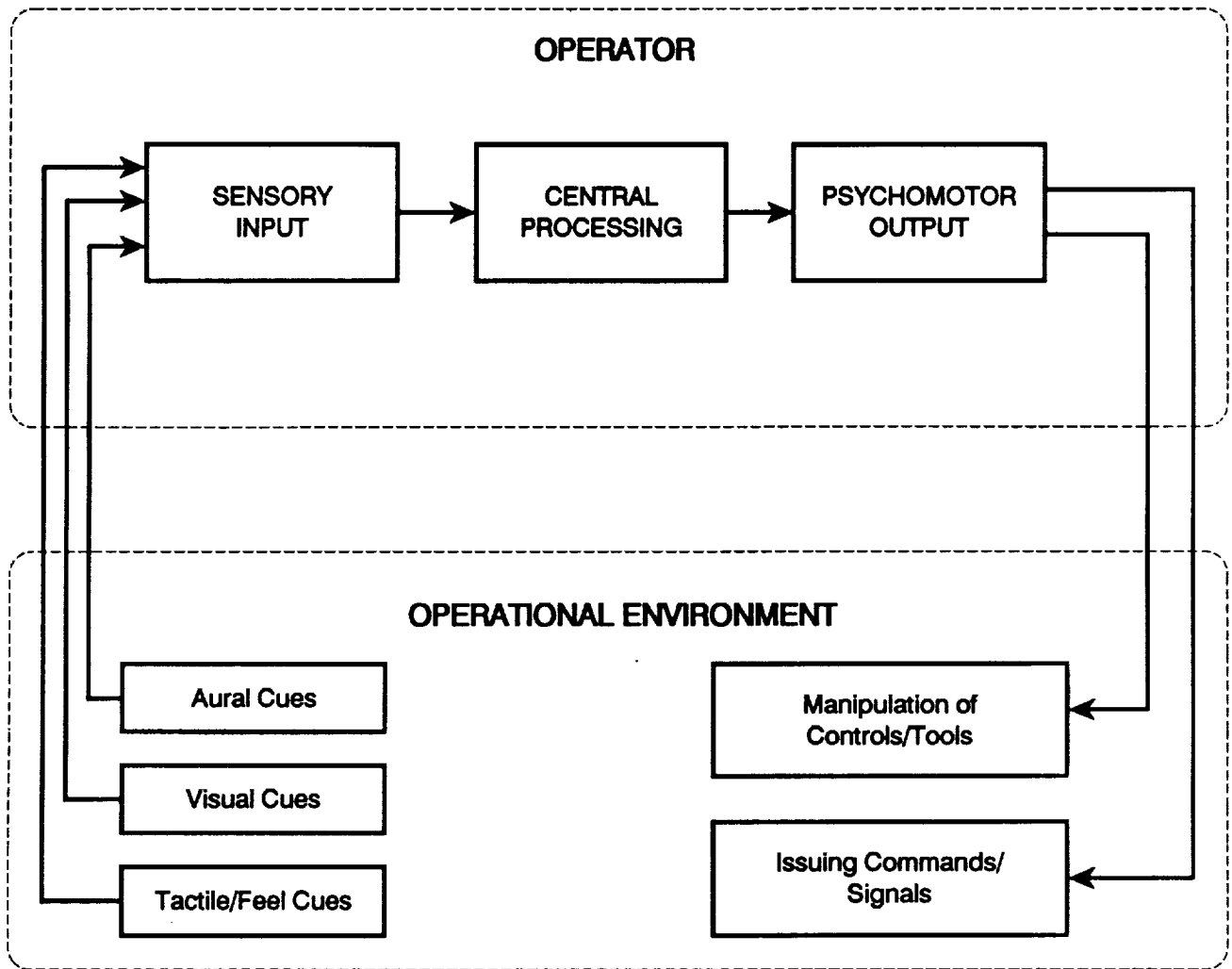
Information Processing Demands:

ANALYZE THE TASKS ASSOCIATED WITH EACH OBJECTIVE FOR THE INFORMATION PROCESSING DEMANDS THEY PLACE ON THE TRAINING ENVIRONMENT. USE THIS TO REFINE FIDELITY REQUIREMENTS.

The above analyses serve to define, perhaps in a basic way, the features and cues necessary to elicit the desired student behavior to accomplish each hands-on objective. The final analysis step is to refine the basic definitions (if necessary) so that they indicate the degree to which these cues and features must replicate in the simulator the actual job environment. If the preliminary fidelity analysis has yielded sufficient results, further definition may not be necessary, and this final step may serve only as a check on the fidelity decisions made. In any case, the desired end product is a requirements definition for each objective which specifies the elements needed for training, and the accuracy with which they must be provided while leaving the designer free to choose the manner of implementation.

In order to evaluate the degree of fidelity required of the simulated experiment to train each objective, it is helpful to analyze the tasks associated with each objective in the context of the information processing demands of the operational setting. From this perspective, the human operator is perceived as performing primarily an information processing function in accomplishing each task. The demands made on the operator by the operational environment during the accomplishment of an objective can be viewed (Figure 3-8) as a sequential flow of three information processing stages:

- a) The sensory input stage refers to the period during which the operator obtains the information needed to correctly accomplish an objective.
- b) The central processing stage is when cognitive skills and strategies are employed in order to determine the correct action in response to the sensory inputs.
- c) The psychomotor output stage of the objective refers to the time when the desired behavioral response to input stimuli is performed.



**Figure 3-8. Accomplishment of a Task/Objective
in Terms of Its Information-Processing Demands**

For each information processing stage involved in the accomplishment of an objective, the required fidelity of the cues and features necessary for task performance can be gauged by the CFD rating for that task. CFD (Task Criticality, Frequency, Difficulty) ratings are subjective evaluations made during Task Analysis. For the sensory input stage, low CFDs indicate a lesser dependency on physical fidelity for processing the information needed to perform a task. Higher CFDs on the other hand, reflect the need for greater physical fidelity in the training device features providing sensory inputs. Similarly, for tasks involving psychomotor output, high CFDs indicate the need for a greater degree of physical fidelity in the controls used in the operational setting since there is greater dependency on control characteristics during the expression of the behavioral response. An example of this would be the flight controls on a cockpit procedures trainer versus a full flight simulator. Since the control tasks taught in a procedures trainer probably will not involve actually piloting the aircraft, the controls may be quite rudimentary. In a full flight simulator, however, where students are required to acquire dynamic interactive flying skills, the controls must be accurate with regards to mass, damping, spring constants, and many other characteristics of the actual flight controls.

CLASSIFY THE TRAINING TASKS ASSOCIATED WITH EACH OBJECTIVE
ACCORDING TO THE TYPE OF LEARNING WHICH EACH REPRESENTS

The first step in determining specific fidelity levels for each objective is to map the training tasks for each objective onto basic learning tasks. This will enable the information processing demands placed by the operational environment during the accomplishment of an objective to be seen more clearly. Table 3-7 lists and explains 11 elemental learning tasks which should encompass the range of activities (tasks) involved in payload operations. Once a task has been classified, Table 3-8 can be used to provide insight into the typical focus of each task in terms of the information processing stage(s) where fidelity determinations must be made.

Table 3-7. Eleven Types of Elemental Learning Tasks

<u>Characteristics of Training Objectives Within Task Categories</u>			
Names of Learning Tasks	Action Verbs	Behavioral Attributes	Examples
1. Recalling Bodies of Knowledge	Answer Define Express Inform Select	<ol style="list-style-type: none"> Concerns verbal or symbolic learning. Concerns acquisition and long-term maintenance of knowledge so that it can be recalled 	<ol style="list-style-type: none"> Recalling equipment nomenclature or functions. Recalling system functions, such as the complex relations between the system's input and output. Recalling physical laws, such as Ohm's law. Recalling specific radio frequencies and other discrete facts.
2. Using Verbal Information	Apply Arrange Choose Compare Determine	<ol style="list-style-type: none"> Concerns the practical application of information. Generally follows the initial learning of information through the use of the guidelines for recalling Bodies of Knowledge. Limited uncertainty of outcome. Usually little thought of other alternatives. 	<ol style="list-style-type: none"> Based on academic knowledge, determine which equipment to use for a specific real world task. Based on an academic knowledge of the system, compare alternative modes of operation of a piece of equipment and determine the appropriate mode for a specific real work situation. Based on memorized knowledge of radio frequencies, choose the correct frequency in a specific real world situation.
3. Rule Learning and Using	Choose Conclude Deduce Predict Propose Select Specify	<ol style="list-style-type: none"> Choosing a course of action based on applying known rules. Frequently involves "if . . . then" situations. The rules are not questioned; the decision focuses on whether the correct rule is being applied. 	<ol style="list-style-type: none"> Applying the "rules of the road." Solving mathematical equations (both choosing the correct equation and the mechanics of solving the equation). Carrying out military protocol. Selecting proper fire extinguisher for different type fires. Using correct grammar in novel situation, covered by rules.

Table 3-7. (Continued)

<u>Characteristics of Training Objectives Within Task Categories</u>			
Names of Learning Tasks	Action Verbs	Behavioral Attributes	Examples
4. Making Decisions	Choose Design Diagnose Develop Evaluate Forecast Formulate Organize Select	<ol style="list-style-type: none"> 1. Choosing a course of action when alternatives are unspecified or unknown. 2. A successful course of action is not readily apparent. 3. The penalties for unsuccessful courses of action are not readily apparent. 4. The relative value of possible decisions must be considered - including possible trade-offs. 5. Frequently involves forced decision-making in a short period of time with soft information. 	<ol style="list-style-type: none"> 1. Choosing frequencies to search in an ECM search plan. 2. Choosing torpedo settings during a torpedo attack. 3. Assigning weapons based on threat evaluation. 4. Choosing tactics in combat - wide range of options. 5. Choosing a diagnostic strategy in dealing with a malfunction in a complex piece of equipment. 6. Choosing to abort or commit oneself to land during the critical point in the glidepath.
5. Detecting	Detect Distinguish Monitor	<ol style="list-style-type: none"> 1. Vigilance - detect a few cues embedded in a large block of time. 2. Low threshold cues; signal to noise ratio may be very low; early awareness of small cues. 3. Scan for a wide range of cues for a given "target" and for different types of "targets." 	<ol style="list-style-type: none"> 1. Detecting sonar returns from a submarine target. 2. Visually detecting the periscope of a snorkeling submarine during daytime operations in a sea state of three. 3. Detecting, through a slight change in sound, a bearing starting to burn out in a power generator.
6. Classifying	Identify Recognize Differentiate Classify	<ol style="list-style-type: none"> 1. Pattern recognition approach of identification - <u>not problem solving</u>. 2. Classification of nonverbal characteristics. 3. Status determination - ready to start. 4. Object to be classified can be viewed from many perspectives or in many forms. 	<ol style="list-style-type: none"> 1. Classifying a sonar target as "sub" or "non-sub." 2. Visually classifying a flying aircraft as "friend" or "enemy" or as a specific aircraft type. 3. Determining that an identified noise is a wheel bearing failure, not a water pump failure, by rating the quality of the noise - not by the problem solving approach.

Table 3-7. (Continued)

<u>Characteristics of Training Objectives Within Task Categories</u>			
Names of Learning Tasks	Action Verbs	Behavioral Attributes	Examples
7. Identifying Symbols	Identify Read Transcribe	<ol style="list-style-type: none"> 1. Involves the recognition of symbols. 2. Symbols to be identified typically are of low meaningfulness to untrained persons. 3. Identification, not interpretation, is emphasized. 4. Involves storing queues of symbolic information and related meanings. 	<ol style="list-style-type: none"> 1. Reading electronic symbols on a schematic drawing. 2. Identifying map symbols. 3. Reading and transcribing symbols on a tactical status board. 4. Identifying symbols on a weather map.
8. Voice Communicating	Advise Answer Communi- cate Converse Direct Express Instruct Interview List Order Report Speak	<ol style="list-style-type: none"> 1. Speaking and listening in specialized terse language. 2. Often involves the use of a specific message model. 3. Also concerns clarity of voice, enunciation, and speed. 4. Timing of verbalization is usually critical - when to pass information. 5. Typically characterized by redundancy in terms of information content. 6. Involves extensive use of previously overlearned verbal skills, or overcoming overlearned interfering patterns. 7. Task may be difficult due to presence of background noise. 	<ol style="list-style-type: none"> 1. Officer giving oral orders and receiving reports. 2. Sonar operator passing oral information over communication net. 3. Instructions by GCA operator to pilot in landing aircraft.

Table 3-7. (Continued)

<u>Characteristics of Training Objectives Within Task Categories</u>			
Names of Learning Tasks	Action Verbs	Behavioral Attributes	Examples
9. Recalling Procedures, Positioning Movement	Activate Adjust Align Assemble Calibrate Disassemble Inspect Operate Service	<ol style="list-style-type: none"> 1. Concerns the chaining or sequencing of events. 2. Includes both the cognitive and motor aspects of equipment set-up and operating procedures. 3. Procedural check lists are frequently used as job aids. 	<ol style="list-style-type: none"> 1. Recalling equipment assembly and disassembly procedures. 2. Recalling the operation and check out procedures for a piece of equipment (cockpit check lists). 3. Following equipment turn-on procedures - emphasis on motor behavior.
10. Guiding and Steering, Continuous Movement	Control Guide Maneuver Regulate Steer Track	<ol style="list-style-type: none"> 1. Tracking, dynamic control: a perceptual-motor skill involving continuous pursuit of a target or keeping dials at a certain reading such as maintaining constant turn rates, etc. 2. Compensatory movements based on feedback from displays. 3. Skill in tracking requires smooth muscle coordination patterns - lack of overcontrol. 4. Involves estimating changes in positions, velocities, accelerations, etc. 5. Involves knowledge of display - control relationships. 	<ol style="list-style-type: none"> 1. Submarine bow and stern plane operators maintaining a constant course, or making changes in course or depth. 2. Tank driver following a road. 3. Sonar operator keeping the cursor on a sonar target. 4. Air-to-air gunnery - target tracking. 5. Aircraft piloting such as visually following a ground path. 6. Helmsman holding a course with gyro or magnetic compass.

Table 3-7. (Continued)

<u>Characteristics of Training Objectives Within Task Categories</u>			
Names of Learning Tasks	Action Verbs	Behavioral Attributes	Examples
11. Performing Gross Motor Skills	Carry Creep Fall Jump Lift Run Swim Throw	<ol style="list-style-type: none"> 1. Perceptual-motor behavior - emphasis on motor. Premium on manual dexterity, occasionally strength and endurance. 2. Repetitive mechanical skill. 3. Standardized behavior, little room for variation or innovation. 4. Automatic behavior - low level of attention is required in skilled operator. Kinesthetic cues dominate control of behavior. 5. Fatigue or boredom may become a factor when skills are performed over an extended period of time or at a rapid rate. 6. Fine tolerances. 	<ol style="list-style-type: none"> 1. From a kneeling position, throw an M67 Fragmentation hand grenade 40 meters on target within Effective Casualty Radius (ECR) using acceptable technique. 2. Wearing a utility jacket, utility trousers, combat boots, and armed with an M16 rifle, traverse 75 meters in deep water using correct form. 3. Demonstrate the proper techniques for a Parachute Landing Fall (PLF) in open terrain. 4. Demonstrate the proper technique of creeping at night across open terrain with a rifle. 5. Demonstrate the proper techniques of chin-ups starting from 'dead' hand, palms toward face position.

LEARNING TASK	TRAINING OBJECTIVE STAGE		
	Sensory Input	Central Processing	Psychomotor Output
1. Recalling bodies of knowledge	Low	Med	Low
2. Using verbal information	Low	Med	Low
3. Rule learning and using	Low	High	Low
4. Making decisions	Low	High	Low
5. Detecting	High	High	Low
6. Classifying	High	High	Low
7. Identifying symbols	Med	Med	Low
8. Voice communicating	High	High	Low
9. Recalling procedures positioning movement	Low	Med	Med
10. Guiding and steering, continuous movement	High	Med	High
11. Performing gross motor skills	Low	Low	High

Table 3-8. CFD Rating Potential for Each Information-Processing Stage by Learning Task

ASSIGN EACH TASK TO ONE OR MORE OF THE INFORMATION
PROCESSING STAGES OF THE OBJECTIVE

DETERMINE THE FIDELITY LEVEL REQUIREMENTS OF EACH TASK
BY CONSIDERATION OF THEIR INDIVIDUAL CFD RATINGS

An objective might be comprised of only one task, spanning all three information processing stages, or multiple tasks, in which case each task might have a significant relevance to only one stage of the information processing sequence. As an example, consider Table 3-8 which illustrates typical CFD distributions for various types of learning tasks. Note that "Rule Learning and Using" holds a "High" CFD rating only in the Central Processing stage. Sensory Input and Psychomotor Output ratings are low, indicating that this particular learning task by itself creates little need for accurate input or output sensory cues. An objective comprised of this one task would place little demands on the display and control fidelity of an experiment simulator. On the other hand, if additional tasks were included in this objective, they would also be analyzed for CFD distribution and could well necessitate greater fidelity. In any case, consideration of the entire objective as an information processing activity should help to indicate the fidelity level requirements for each feature of the training environment.

In most cases, the CFD rating determined for each task during Task Analysis will be appropriate for, and can be used to analyze the information processing stage or stages to which the task has most relevance. For example, a "Gross Motor Skill" type task has most information processing activity focused in the psychomotor Output stage. Therefore, the task's CFD rating will have much more significance to fidelity or instructional strategy decision making in that stage than in the other two.

EXAMPLE

The sensory input stage refers to the period during which the operator obtains the information needed to correctly accomplish an objective. A Xerox machine operator for example, to perform a task such as to detect a certain malfunction, might have available a variety of input stimuli such as aural cues, malfunction codes, and LED indicators which would alert him or her to the presence of a problem. If the operator is dependent upon these cues to detect the malfunction, the CFD rating associated with comprehension of the cues will be an indicator of the degree of fidelity required. For example, if a particular stimuli occurs rarely, is of minor importance to task

accomplishment, and is easily noticeable, its CFD rating for comprehension by the operator will be low and would not need to be represented with great accuracy. For stimuli with high CFD ratings, a greater degree of fidelity to the operational environment would be necessary.

The central processing stage is when cognitive skills and strategies are employed in order to determine the correct action in response to the sensory inputs. It is at this point, that the Xerox machine operator would evaluate the sensory inputs to diagnose the malfunction, and decide what to do about it. High CFD ratings for this task (diagnose malfunction), would be appropriate if malfunction diagnosis (learning task - Classifying) in this case was critical to objective success, was a frequently occurring activity, and/or was difficult to do correctly.

For these central cognitive processing activities, however, the CFD ratings carry a somewhat different meaning than for the other processing stages. A high rating indicates a greater dependency on cognitive skills and operational strategies by the student to perform adequately. Since these central cognitive processes are internal to the operator, they indicate a need for greater feedback cues, rather than environmental fidelity, to help develop cognitive skills and strategies, and heighten training effectiveness in those areas.

This is a fidelity consideration only to the extent that accurate feedback cues actually exist in the operational setting. If more feedback is desired than can be provided by the task environment, artificial means such as instructor comment or performance testing may be used which relate more to the realm of training techniques than they do to physical fidelity. Thus, CFD ratings on central processing type tasks tend to guide instructional strategies rather than indicate necessary levels of physical correspondence of the training device to the operational setting.

The psychomotor output stage refers to the time when the desired behavioral response to input stimuli occurs. For the Xerox machine operator, this is when he or she actually repairs or removes the malfunction. This task could be classified as a Recalling Procedures, Positioning Movement type and would carry a high CFD rating if, (like the above central processing task) it was critical to objective success, performed often, and/or was difficult to do. Judgement however, is an important factor when interpreting CFD ratings. If, for example the malfunction must be repaired to accomplish the objective, then obviously the repair task is Critical. On the other hand, if the malfunction occurs rarely, or is extremely easy to resolve, then good judgement would preclude great efforts made to provide high fidelity.

Whereas during initial fidelity analysis phases, cues and features might have been specified without regard to the exact fidelity level at which they would be presented, the CFD parameters allow one to gauge the level needed in the simulator to train the task. Use the results garnered in this step to modify or add to the trainer functional requirements already determined. In most cases, this should involve a more detailed and specific delineation of specific performance parameters, but in some cases could also result in modifications to basic trainer functionality. As always, rationale for every decision made should become part of the evaluation results recorded in the Experiment Database.

The learning taxonomies discussed in this methodology are not unique, and in fact many different classification techniques for learning tasks etc. have been formulated and would serve as guides for fidelity determinations. By working through the methodology outlined here, it should be clear that they are principally aids for educated common sense decisions, rather than infallible analysis tools. Evolution of this methodology must come through application of the results of empirical research, with the end goal of maximum transfer of training. This can be accomplished by systematically relating measured training effectivity to specific levels of fidelity, under various conditions, with the aim of determining the minimum fidelity required to enable training specific tasks.

The following general rules should be considered when constructing hands-on media fidelity or functional requirements:

- a) Start with zero. Do not assume any function, capability, or aspect of an experiment is necessary unless justified by a training requirement.
- b) Relate each function or performance parameter chosen for an experiment simulator to the requirements implied by specific task and objective information.
- c) Start with simple, low fidelity approaches to meeting training requirements, only adding complexity or high fidelity as it is required to improve training.
- d) In general, fidelity should increase as task complexity and student proficiency increase, but only within the bounds delimited by training objectives.

3.2.2 Hands-On Media Functional Specifications

ANALYZE THE COMMON FUNCTIONAL REQUIREMENTS AMONG THE
HANDS-ON OBJECTIVES TO DEFINE TRAINING DEVICE CATEGORIES

DEVELOP FUNCTIONAL SPECIFICATIONS FOR EACH CATEGORY OF
HANDS-ON MEDIA

Once the functional requirements for each hands-on training objective and candidate training medium have been defined, they are examined collectively to establish simulator categories based on similar requirements. These trainer categories are established on the basis of the media candidates for each objective, stage of training, overall instructional strategy, and level of fidelity required. The output of this step shall be collective functional characteristics which will serve to define various levels of hands-on media fidelity or functionality. Functional Specifications are then developed for each of the required hands-on media.

Not all objectives, however, will relate directly to single experiment simulators and so cannot be grouped with other objectives in a Functional Specification. These include Mission and Science level objectives which are concerned with the operation of multiple experiment simulators to train teamwork, timeline, and protocol skills among the ground and flight crews. These objectives will be input to Syllabi Development so that they may receive consideration in the total training curriculum. They will also be addressed by the Simulator Requirements Derivation activity when the overall training plan for each increment is coordinated.

When assembling the Functional Specification for a training device, it is desirable to define the device in terms of the tasks it must be capable of training. Specifying device requirements in terms of desired student behavior gives the simulator builder more flexibility, due to the "performance" characteristics of the trainer specification. The goal is to tell the designer what the simulator must do, while not constraining him or her as to how the simulator should do it. The specification should include:

- a) A list of all objectives to be trained including their tasks, subtasks, and sequences of performance. This should be output on command from an automated database utility.

- b) A complete description of the S-R (Stimulus-Response) conditions for each control and display for each separate task.
- c) A description of each task's initiating and terminating conditions, actions (behavior) required, and relevant controls and displays.
- d) A description of all conditions for task execution such as constraints, relevant contingencies, malfunctions, and performance standards.
- e) A description of the degree of fidelity necessary for each cue, including required functionality as well as quality of simulation (tolerances).
- f) Any hardware constraints or decisions made for reasons extraneous to the requirements analysis activity.

Since the specification will consist of a summary of requirements for a group of objectives, it is possible that the requirements may conflict at points. It will be the simulator requirements developer's task to resolve these contradictions in the most cost-effective manner when the finalized hands-on media Functional Specification is assembled. This will be discussed further in Simulator Requirements Derivation.

At the completion of functional and fidelity specification for an experiment's training devices, the customer (PI) associated with each experiment should be asked to verify all aspects of the fidelity and functional analysis. This involves tracing the audit trail from reformatted task and objective data to functional requirements and final device specification, with examination of recorded rationales for each decision. It should be possible to justify each training feature against specific behavioral training objectives.

3.2.3 Simulator Functional Requirements and Functional Specifications Procedures Summary

Derive simulator functional requirements for each hands-on media objective based on:

- Task Analysis Data
- Lesson Specifications
- Empirical Studies.

Organize functional requirements for each objective on Simulator Functional Requirements Forms.

Refine physical and functional fidelity requirements through consideration of the information processing demands of each objective.

Group functional requirements into training device categories.

Summarize functional requirements into hands-on media Functional Specifications.

The output of media and methods selection will be the following:

- a) A recommended medium and methods of instruction for each objective.
- b) Functional Specifications for both hands-on media as well as for academic media using hardware, software, or courseware. These specifications are developed during media selection when media are not only chosen, but also characterized as to their adequacy to handle an acceptable number of instructional requirements relative to each assigned objective.

The media and methods recommendations for both academic and hands-on media will be used for further instructional planning in the Syllabi Development activity. The functional specifications for the hands-on media will be input to the Simulator Definition Analysis activity (see Figure 3-3).

3.3 Syllabi Development

With the establishment of candidate methods and media for each training objective, and the development of media functional specifications, the active learning environment should be well defined. At this point then, the basic learning structure may be detailed as to the content and organization of the curriculum. Objectives are clustered into lessons, and sequenced within each lesson to optimize skill and knowledge acquisition. Lesson specifications are written, documenting instructional breadth, depth, methods, and media for subsequent development. Separate training tracks are established for each crew position (for example, Mission Specialist) from sequences of lessons. Figure 3-9 illustrates the Syllabi Development process.

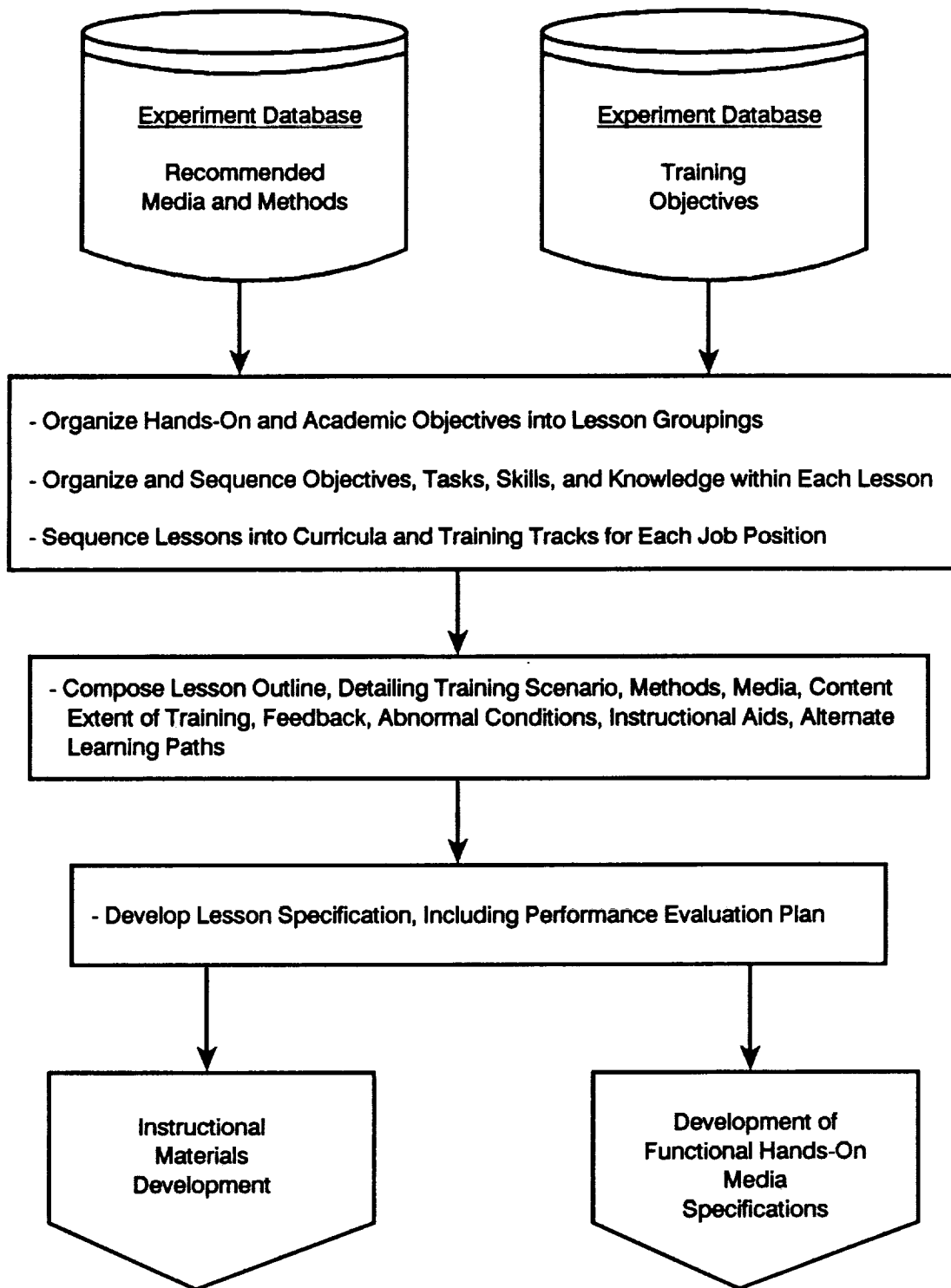


Figure 3-9. Syllabi Development

3.3.1 Lesson Organization

ORGANIZE HANDS-ON AND ACADEMIC OBJECTIVES INTO LESSON GROUPINGS

ORGANIZE AND SEQUENCE OBJECTIVES, TASKS, SKILLS, AND KNOWLEDGE WITHIN EACH LESSON.

SEQUENCE LESSONS INTO CURRICULA AND TRAINING TRACKS FOR EACH JOB POSITION

Lessons are composed of sets of objectives, which in turn can be both subsets and supersets of the tasks to be trained. Internal and external lesson organization therefore, effects the arrangement of objectives and tasks within a training sequence, and defines the structure of the instructional system. This structure determines to a great extent, the way in which relationships between job elements are seen and understood. The instructional content therefore, of a training course may be made more meaningful by appropriate lesson ordering. In addition, gaps in training, and duplication of training may be avoided, and an orderly building of skills and knowledge may be facilitated.

In addition, to fully accommodate students' individual differences and their intended assignments, the instruction can be modularized into separate segments, each covering one or more objectives. Individual programs would be assembled for each student, based on their prior experience, training and job assignment. The result would be separate training tracks for each position and within each track, unique courses of instruction fitted to individual needs. Structuring courses in this way will eliminate unneeded instruction and reduce average course length.

The differences in education, experience and possibly, aptitudes between the two groups suggests that a two-tier course system may be necessary for selected topics (such as experiment familiarization) in which the flight and ground crews must both be trained. This partial duplication of effort can be mitigated by flexibility in the instructional materials (as described in the preceding sections) so that one set of developed courseware can be used by both groups. Experiment familiarization courses for example which both flight and ground crews might be expected to need could be geared to the perceived abilities of the ground controllers, but presented in a flexible manner, so that the flight crew could use the same courseware. Even if different facets of the same topic (such as experiment operations) were to be emphasized for each group, the differences could be

modularized within each curriculum. Since classroom training does not lend itself to this kind of flexibility, it may be wise to limit the use of classroom training to courses unique to each group. Obviously there are tradeoffs to be considered, such as the cost-effectivity of twin classroom training tracks on the same topic versus flexible self-study materials or CBT courseware.

A Terminal Objective can be broken down into the component Enabling Objectives and basic skills and knowledge needed to accomplish certain tasks. The instructional designer has much latitude in how the objectives and the tasks which they represent may be organized in order to reach the training objectives. For much of payload training, sequencing of the tasks for each objective will often follow the job order since so much of the training is procedural in nature. However, course content can be sequenced in a variety of ways. These various methods should be considered when organizing a lesson, to optimize training effectiveness:

Traditional Order: With the traditional approach, tasks are clustered together which are highly related in some way. These ways could include tasks which are highly interactive, or which are alternative methods to reach the same objective. A variant of this approach is to break down tasks into subtasks with limited objectives and arrange them so as to perform the easier tasks first. The skill and knowledge requirements to perform each task would increase progressively through the task sequence. Another variant of this philosophy clusters tasks on the basis of commonalities between the task actions or what they act upon.

Component Ordering: The Component Ordering approach concentrates first on the component skills and knowledge supporting the tasks and subtasks within an objective. Training for proficiency in those skills and knowledges would be accomplished before using them to perform a task. Also, the skills and knowledges from different tasks and even different objectives could be trained together on the basis of their similarity or their relevance to a single system function. Evidence exists, however, (Schneider, W. 1985) to suggest that training on a component skill should be interspersed with training on other skills used to accomplish the same task, to facilitate perception of their interrelationships.

Learning Type Order: This approach clusters tasks based on the types of learning involved in mastering them. Thus, all tasks requiring association type learning for example, would be grouped together. Subgroupings would then be formed of tasks which relate to common equipment (such as an instrument), cue type (such as a CRT), or response type (such as a keyboard).

Job Performance Order: In this method, instructional content is presented in the order in which tasks and task elements are performed on the job. This is the most straightforward arrangement method and has the advantages of providing the most realism and maximization of learning transfer from the training environment to the job situation. In addition, the transfer of skill and knowledge elements from one task to another related task is greatly facilitated. This method is typically used for teaching sequences made up of a number of fixed steps, such as simulator training on experimental procedures. It should find a great deal of application for ground and flight payload operations training with mockups and indeed, is such an intuitively obvious approach for procedures training that it is easily the most widely used technique in simulator training. It should be noted, however, that alternate methods exist (as discussed below) and, especially for introductory level instruction, alternate approaches may offer more effective solutions.

Psychological Order: This method is based on the principle that complex tasks and ideas can be more easily learned by first understanding their component tasks and concepts. Under this approach, the instructional presentation sequence is arranged such that the learner is taken from the simple to the complex, the concrete to the abstract, or from the general to the specific. At the start of any instructional sequence for example, student motivation may be increased by first relating the introductory instruction to what the student already knows about the topic, and from there, to progressively present more difficult material in a building block fashion.

An introductory experiment briefing for example, could begin with a summation of what the students learned at the PI facility, proceeded by relating that information to the general principles on which the experiment is based, then applying those principles to a specific part of the experiment. Similarly, component experiment functions might first be discussed separately before explanation of their integrated operation. This provides for a smooth progression and buildup of experiment operating principles which is usually not possible when following the Job Performance Order technique, because the performance of an actual job typically involves a series of tasks of random difficulty levels.

Instruction on complex motor tasks is generally more effective when initiated with the practice of simpler, transferable tasks. Training of substrate handling for the Membrane Production Facility for example, could begin by using the sample manipulation devices to perform simple subsets of typical maneuvers before attempting a coordinated sequence of handling operations.

Logical Order: For greater learning effectiveness, the progression of instruction should normally move from simple to complex knowledges and skills. Often however, it is also desirable to preserve the ordering of tasks as they are required for the actual job. An example would be this would be the case of simulator training for the operation of a complex experiment. For realism, the experiment operations should be practiced in a job ordered sequence. This sequence, however, may not be optimal for learning, as it may cause the most difficult tasks to be performed before the easier ones. One solution is to break the difficult tasks into their basic elements, and practice them separately prior to training for the entire job (provide more training). For greatest efficiency though, a better solution could be to arrange the learning session so that the student performs the easier tasks and observes the instructor performing the more difficult ones. Later, the entire sequence is performed by the student with lesser degrees of instructor assistance. Another variation would be to first have the instructor demonstrate the entire experiment process and then perform a step-by-step demonstration with step-by-step student participation; then recombine the job elements for a complete runthrough. In any case, the idea is to provide a progressive approach to building the desired skills and knowledge, while preserving the realism necessary to effective transfer of learning.

These many alternatives are not meant to suggest that efficient objective, task, skill, and knowledge ordering need be an arduous undertaking. Rather, this process should be clear in most cases. The above possibilities are listed solely to enlighten the reader as to the range of possibilities. All ordering methods could potentially have a place in payload operations training. The choice of methods used therefore, should be determined by the nature of the skill or knowledge being taught as well as the training medium. The following general sequencing rules should be considered:

- a) Place easily learned objectives early in the sequence. Place complex and cumulative skills later.
- b) Introduce concepts at the first point where understanding of those concepts is necessary for successful performance.
- c) Introduce instruction on skills before the point where they will be combined with other skills and used.
- d) Where possible, train procedural skills and knowledges in the same order as required on the job.

e) Introduce a knowledge or skill in the context (task) where it is most likely or most frequently to be used.

f) Provide for practice of skills and concepts in areas where the transfer of like or related skills from other tasks is not apt to occur.

3.3.2 Instructional Requirements

DETERMINE THE INSTRUCTIONAL REQUIREMENTS FOR EACH OBJECTIVE, AS AN INPUT TO LESSON DESIGN

A central part of Syllabi Development must be to determine the instruction necessary to fulfill the objectives and tasks around which the lessons are designed. These instructional requirements will be defined through analysis of task attributes, and considering the initial skills and knowledges of the incoming trainees.

The skills and knowledges identified in the Objectives Hierarchy, along with the Tasks, Task attributes and prevailing training policy will be used to help determine the advisability of providing training, the extent and amount of training required, and the training method. Task attributes which should be considered in determining these Instructional Requirements include:

a) **The Number and Kind of People Who Will Perform the Task:** This will help determine who will receive training and also the training method. For example, if a task is to be performed by a very limited group, it might be excluded from classroom and CBT training and only offered to the necessary trainees during simulator exercises.

b) **Task Criticality:** In determining the Instructional Requirement for a task, criticality is sometimes more important than how many people perform it or how often it must be done. For example, a task which is done infrequently by one person may not seem to merit much attention. If however, the mission were to be severely impaired if the task were not accomplished, then it would make sense to provide thorough training for that task and to provide that training to more than one trainee.

c) **Frequency of Performance:** A task often done gives a greater chance to develop proficiency. This could indicate a

limited degree of instruction followed by On-the-Job Training (OJT). Conversely, a task seldom done, but highly critical might require a high degree of initial training.

d) Learning Difficulty: Easy tasks which can be learned "by doing" will require little formal instruction. Difficult tasks for which OJT is not feasible or desirable will require a high degree of formal instruction.

e) Time Interval Before First Performance: If there will be a long interval between when a task is trained, and when it will be performed, some remedies should be planned. Options include deferring training until later in the program, increasing training to aid student retention, or planning refresher training, before or during the Mission.

f) Task Commonality: If a Task is performed as part of other procedures in other Task Hierarchies, the total training for that task must be calculated. It may well be that the other procedures will provide sufficient training for a particular Task.

g) Marshall and SSF Training Policies: The prevailing training philosophy must be considered for all training decisions. This will affect resource utilization, degree of cross-training, amount of OJT, training approach, and decision parameters such as numbers of trainees to justify CBT development.

In determining the Instructional Requirements, the above factors (and any others) must be considered collectively, as well as individually. An objective for example, based upon a task which has been classified as Mission-critical but which is both easy to perform and previously taught at the PI site, might still require refresher training, albeit to a reduced degree. On the other hand, if a task consists of operating a SS system and has been trained elsewhere, it may be assumed that the incoming student will possess the requisite skill or knowledge. In any case, a balanced decision must be made, taking into account all of the task attributes, as well as the prevailing MSFC and SSF Program training policies and guidelines.

Since the methods used will be influenced by resource availability, schedule and cost constraints, this is an appropriate time for preliminary identification of resource requirements such as manpower and system-peculiar or long leadtime training equipment. Total resource requirements will be identified when the total instructional program is defined.

3.3.3 Lesson Design

COMPOSE LESSON OUTLINE, DESCRIBING TRAINING SCENARIO,
METHODS, MEDIA, CONTENT, EXTENT OF TRAINING, FEEDBACK,
ABNORMAL CONDITIONS, INSTRUCTIONAL AIDS, ALTERNATE
LEARNING PATHS

Lessons are outlined for each subject matter topic, covering one or more training objectives. Designing the lessons around the objectives ensures that they will be focused on the activities required to demonstrate that the desired learning has taken place. The coverage of each lesson should be managed in order to encompass enough material to result in a significant learning accomplishment, yet be restricted to a single subject matter topic. As a rule of thumb, an academic training session should be limited to one hour in length, and a hands-on training session to three hours in length.

Depth of Training: In general, the extent of training accorded a subject should be a function of its criticality and learning difficulty. However, when determining how much to provide, it is usually better to give minimal instruction on the first design iteration and allow the instructional validation process to show where more is needed. Use of instructional validation as well as feedback from students, instructors and performance evaluations will allow the amount of instruction to be optimized. If more instruction is provided in the beginning than necessary though, the training will be unnecessarily expensive, and there will be no way to detect that this is so.

Environmental Options: To increase training effectiveness, the lesson designer may, without altering the training medium, make the instructional environment unlike the job environment. This can be done by:

- a) Using feedback to give the students more knowledge of the results of their activities than they would normally receive on the job
- b) Providing training scenes that are appropriate.

For both hands-on and academic instruction, the principle of confirmation (knowledge of results) is an important factor for training efficiency. Informing students of the correctness or incorrectness of their responses enables concentration on points of deficiency, and provides a source for confidence building and motivation. Confirmation can be provided immediately following a

student's response (for example, by instructor comment or automatic feedback from a CBT terminal) or following the completion of a course of instruction (such as test results or performance review). In general, the more immediate the confirmation following a student response, the greater will be its effect on overall learning efficiency. Lessons should be designed to offer more frequent confirmation in the early learning stages to build confidence. Later, feedback can be supplied on a more intermittent basis.

Pacing: The rate at which students progress through an instructional sequence is controlled by course design and media selection. Group-pacing, where all students progress through a course of instruction together, is useful

- a) when the establishment of a group identity is desirable
- b) when the nature of the selected training medium demands it (for example classroom)
- c) when schedule constraints require all students in a group to complete instruction at the same time

This mode is enabled when instruction and subsequent testing is provided simultaneously to all students. Disadvantages with group-pacing are that the quicker students, or those with prior experience, will have to wait for rest of the class, or that slower students will not have an opportunity to attain proficiency. This results in overall training inefficiency.

To optimize training efficiency, it is often possible to allow self-pacing, where students are free to advance at their own rate. With this method, students advance on the basis of their performance in the criterion tests for each lesson. The students who are able to proceed sooner may do so, while the other students each have an opportunity to master each lesson's training objectives.

Learner Characteristics

Appendix A contains an analysis of the probable group characteristics of the incoming trainees for payload operations training. Based on this analysis, general recommendations can be made for payload training lesson design:

Lessons should be designed to allow small, well structured steps, and a slow presentation rate accompanied by a high rate of repetition. Concepts and instructions should be presented in simple language when possible, and the instructional content should be related in a functional context. External feedback and motivation should be supplied via a live instructor or by features intrinsic to the training media and/or instructional materials.

In addition, instructional materials should be flexible, to permit a range of learning rates and the optional repetition of course segments. Specific instruction should be geared to the minimum learning level of the trainees likely to use it, while allowing alternate learning paths for those of greater capability. In the case of CBT courseware for example, learner selected options could be provided to allow branching around auxiliary information, or to proceed immediately to test (for feedback). This kind of flexibility should serve to speed training and maximize resource effectivity, as well as accommodating individual learner characteristics.

In general for payload operations training, given the probable characteristics of the incoming student population, it is recommended that developed instructional plans use a combination of group pacing where necessary, and self paced instruction when possible, in order to optimize training efficiency.

3.3.4 Lesson Specifications

DEVELOP LESSON SPECIFICATIONS, INCLUDING PERFORMANCE
EVALUATION PLAN

The Lesson Specification consists of a detailed outline containing or referencing all information necessary to allow authoring of the actual lesson, and development of enabling instructional materials. Lessons will be developed for both academic and hands-on media. The major input to the Lesson Specification will be the Lesson Outline.

Academic Lesson Specifications: Each specification contains both general lesson information and specific information on each objective covered in the lesson. General information includes a hierarchical "map" of the lesson objectives, a lesson introduction, overall instructional strategy, student prerequisites, and a description of the instructional materials required to conduct the lesson. Specific information on each objective includes the objectives themselves, along with their associated Conditions and Standards of Performance.

Hands-On Lesson Specifications: These are specifications developed for each lesson to be conducted on a trainer or the actual equipment. Each specification contains the elements required for student practice and instructor evaluation of the objectives in the lesson. These consist of the same items as detailed for academic lessons as well as an outline of tasks to

be performed, a description of the instructor guidance to be provided, and references to the academic lessons which support accomplishment of the current objectives.

Performance Evaluation Plan: Both hands-on and academic lesson specifications will include general and specific performance evaluation procedures. These include tests for each objective, as well as Performance Measures for the entire lesson and curriculum. The objective test items measure the specific behavior associated with each objective, and are derived directly from the tests developed during the formulation of the Training Objectives. The Performance Measures are more concerned with overall training effectivity and lesson and curriculum goals. Their derivation must begin with a clear understanding of the various purposes for evaluation and end with a validation of the derived measures against accepted metrics' criteria for each evaluative purpose. This process is discussed in Appendix B, Metrics. Major uses for test items and performance measures include:

- a) Determining the present proficiency or capability of an individual.
- b) Predicting an individual's future performance.
- c) Diagnosing deficiencies and strengths on component processes underlying the skill being acquired.
- d) Determining training effectivity and/or evaluating alternative training methods.

These measures will be used to conduct testing for two principal purposes: Ongoing Simulator Validation and student performance evaluation. Student evaluation results will be used to monitor and adjust training for individual students. Ongoing Validation will feed back recommended changes to either current training or to the training development methodologies.

3.3.5 Instructional Materials Development

The Instructional Materials Development activity receives as input, the Functional Specifications for all academic media, including CBT courseware, Lesson Specifications for both academic and hands-on media, and performance evaluation specifications. Its output consists of CBT courseware, workbooks, tests, charts, study guides, training scripts, films, slides, test plans, and all other materials necessary to support academic and hand-on training. Academic materials will be developed first, while materials for hands-on media will be developed after simulator requirements are delineated at PDR.

At Simulator PDR, the academic instructional materials will be verified for traceability to Instructional Requirements specified in the Instructional Plan. After PDR, with simulator functionality specified, development can proceed for those materials which will directly support the use of experiment simulators for training.

One branch of the Instructional Materials Development activity is responsible for the generation of scripts for the use of instructors during simulator training sessions. These scripts fall roughly into two categories. One kind of script will be designed to fulfill objectives related to individual experiment operation. Another kind of script will be designed to fulfill higher level mission or science objectives, and will teach teamwork and coordinative skills. These include communications protocol, timeline validation, and coordination between flight crew members, and between flight crew members and ground controllers. Both kinds of scripts will be derived from the mission timelines and Crew Activity Plans to fulfill their respective objectives.

Resultant course materials will be presented and reviewed at CDR, in conjunction with designs for the simulators they are intended to support. After CDR, instruction will begin 15 months before launch using the classroom and CBT materials. Experiment simulator materials will see their initial use (and final testing) during Acceptance Verification and Validation when the simulators are used in the execution of training scenarios.

3.3.6 Syllabi Development Procedures Summary

Organize and sequence hands-on and academic objectives, tasks, skills, and knowledges into lesson groupings by consideration of the following methods:

- Traditional Order
- Component Order
- Learning Type Order
- Job Performance Order
- Psychological Order
- Logical Order

Determine instructional requirements for each lesson by consideration of:

- Numbers and Type of Personnel
- Task Criticality, Frequency, and Difficulty
- Task Redundancy
- Marshall and SSF Training Policies

Design Lesson Outlines.

Develop Lesson Specifications and Performance Evaluation Plans.

Develop Instructional Materials Based on Lesson Specifications.

4.0 SIMULATOR REQUIREMENTS DERIVATION METHODOLOGY

Simulator Requirements Derivation is the process whereby detailed simulator hardware and software requirements are produced which reflect Mission and Science as well as individual and integrated experiment training objectives. Its primary inputs consist of PI-provided experiment data and hands-on media Functional Specifications. The process, however, must also take into account overall SS training plans, PTC resources, experiment development schedules, and the planned training curricula for each experiment.

The Training Analysis methodologies (#1 & #2) fulfill the role of Instructional Systems Development (ISD) in producing requirements for complete training systems. Simulator Requirements Derivation (Methodology #3) is a Systems Engineering process designed to utilize these training requirements to formulate simulator requirements. These requirements will in turn, be used as the basis for simulator design and development. It should be noted, that although the outputs from methodologies #1 and #2 provide a major input to Simulator Requirements Derivation, they are not always mandatory. If for example, experiment information was provided too late for a front-end training analysis to be performed, Simulator Requirements Derivation could proceed anyway, with whatever experiment data was available. In the absence of derived training objectives and hands-on media Functional Specifications, the simulator developers would have to make educated assumptions as to the required simulator functionality and fidelity. The simulator would not be as cost-effective, and Verification and Validation activities would be restricted, but satisfactory training would still be possible.

The Simulator Requirements Derivation process is defined here in terms of the data items which will be generated by the developer while deriving simulator requirements. These data items include a) an Experiment Overview Report (EOR), b) a Simulation Approach Document (SAD) for each experiment simulator, c) a description of training scope for each experiment, to coordinate with JSC, d) a Software Top Level Requirements Document (STLRD) for each simulator, and e) a detailed math model and requirements document for each simulator (Experiment Software Requirements Document [ESRD]). Simulator Requirements Derivation, though described here as a sequential process is actually iterative in nature; gradually producing mature simulator requirements as understanding of particular experiments grows and experiment data

becomes available. The process description that follows provides a place for external data and analysis results so that, although analysis is not likely to proceed in the exact order specified, traceability can still be established between data items which will then demonstrate a logical development flow. Figure 4-1 illustrates the overall requirements derivation process.

MAJOR INPUTS

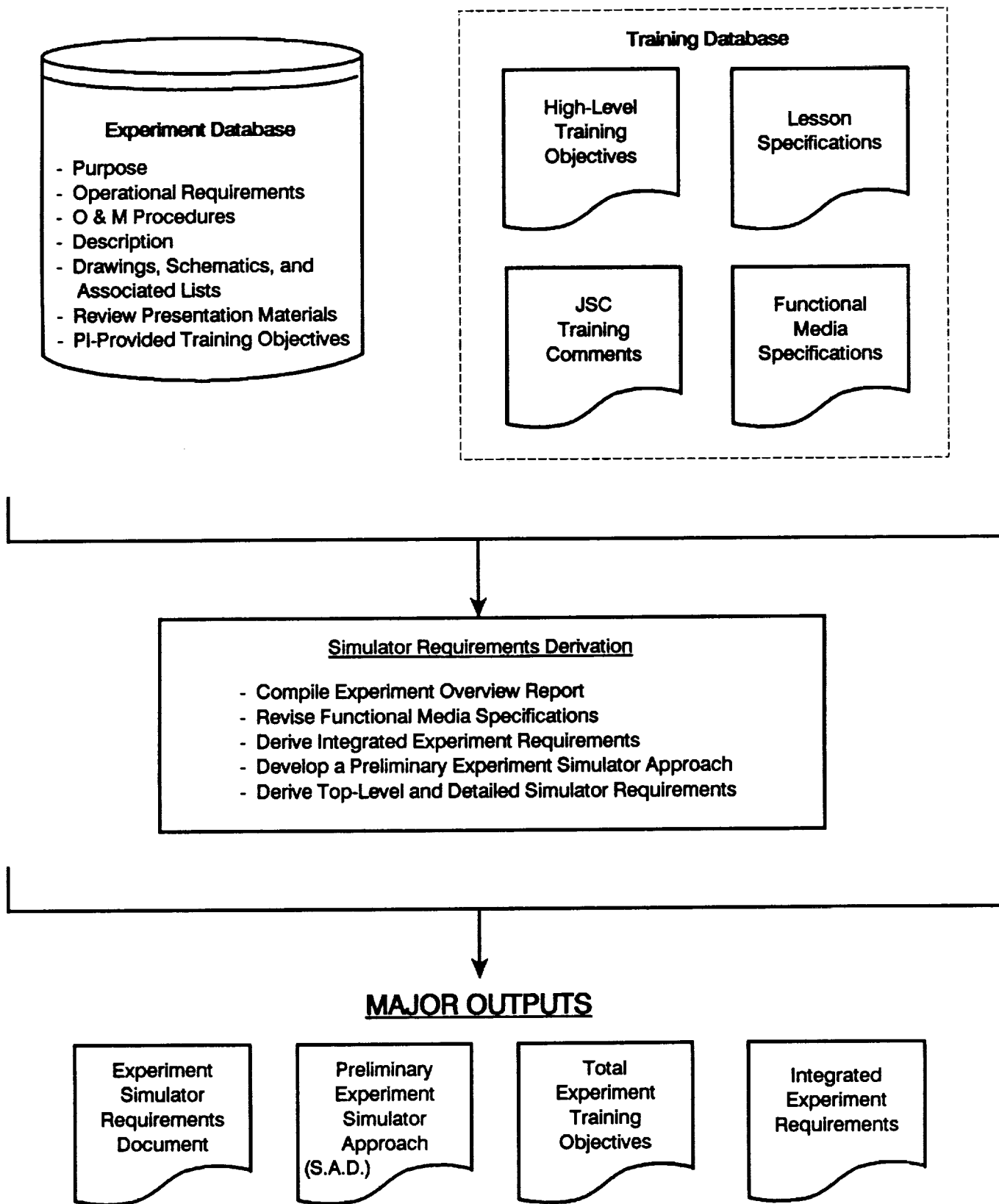


Figure 4-1. Inputs/Outputs of Simulator Requirements Derivation Process

4.1 Experiment Overview Report (EOR)

The EOR represents an initial effort to evaluate an experiment in terms of the simulation and training problems which it represents. Its building blocks are comprised of data items developed as part of the data acquisition phase of the Training Needs Assessment Methodology (#1). These data items have been designed to fulfill the needs of both the Training Analysis and simulator Systems Engineering processes. Therefore, under ideal circumstances, most of the work involved in producing an EOR will already have been done for training analysis, and stored in the Experiment Database. If not, the data items must then be derived from experiment information and stored in the Experiment Database, as described in the procedure for Training Needs Assessment (Section 2.1.1). In addition, if the experiment data has changed or been augmented since the time that the data items were developed, it may be necessary to update them before proceeding with further analysis. Any further data items developed as part of Simulator Requirements Derivation should also be included as part of the database so that all analysis efforts will have access to the same inputs.

4.1.1 Purpose

The EOR is a description of an experiment under development, which can give a sense both of the experiment's complexity and of the problems which may be encountered in its simulation. It will report on the experiment's current status and provide a prognosis for its future development (timeliness of data, availability of hardware, prototypes, simulators, support equipment). This data will be used to help develop an approach to simulating the experiment (hardware vs software, simulation versus stimulation, etc.) for the SAD. In addition, the experiment overview will provide an outline of all available experiment data, including schematics, drawings, conceptual studies, etc. This will serve as an experiment data road map for subsequent analysis efforts, showing what information is available, and where it may be found.

The EOR will also document the experiment training objectives as seen by the PI. These objectives will be considered along with the objectives derived from front-end ISD analysis when the hands-on media Functional Specifications are revised and finalized.

4.1.2 Structure

PRODUCE AN EXPERIMENT OVERVIEW REPORT FROM AVAILABLE
EXPERIMENT INFORMATION

The EOR will utilize the following items from the Experiment Database to compose an experiment overview:

- Experiment Description
- Experiment Purpose
- Experiment Operational Requirements
- Review Materials
- Experiment Operating and Maintenance Procedures
- Experiment Drawings, Schematics and associated Lists, or overview of same (including flowcharts, if available).

The Report will integrate the above information to produce a condensed description of the experiment sufficient to assist developers of the SAD in choosing an overall approach for the experiment simulator(s) and in developing simulator requirements documents. The EOR should be organized as follows:

- a) Experiment Overview
- b) Experiment Status Report
- c) Integrated Experiment Simulator Requirements
- d) PI-Provided Training Objectives
- e) Outline of Available Experiment Data.

Experiment Overview: This section should contain a textual description of the experiment, its science objectives, and major functions. Graphics and text should be used to describe its components, internal and external interfaces, support equipment, and operating controls and displays. Experiment inputs and outputs shall receive special attention, with both a functional description of each parameter as well as a tabular listing of data parameters and their numerical specifications. If convenient, the more detailed input and output information can simply be referenced back to its location in the Experiment Database, rather than being duplicated unnecessarily. The description of experiment inputs and outputs which are included in the EOR should be detailed enough to support the later definition of preliminary simulator requirements.

In addition to experiment inputs and outputs, the section shall also include a description of the data transformations connecting them. This description, which ideally would include both a textual explanation and equations, will aid subsequent efforts concerned with simulation approaches and requirements derivation.

Experiment Status Report:

REPORT ON THE CURRENT STATUS OF EXPERIMENT DEVELOPMENT.
GIVE A FUTURE PROGNOSIS OF THE DEVELOPMENT EFFORT.

The EOR will utilize the Experiment Development Schedule and Experiment Review Materials from the Experiment Database to produce an Experiment Status Report. This report will assess the current development progress of the experiment and provide a prognosis for its continued development. The status report will provide situational data to help make simulator approach decisions. For example, if an experiment has a high probability of being late, or subject to last-minute changes, the recommended simulator design could be slanted toward software simulation (which is easier to change than hardware), or the use of unaltered flight software which could be quickly updated to track experiment changes. Conversely, if experiment development is proceeding apace, it may be possible to accelerate analysis efforts, or start analysis sooner, while other experiments mature. In addition to experiment information, the status report would also discuss any PI-produced simulators-in-progress. This is important information for simulator approach development in determining what simulator and support functions must be supplied by the PTC.

Integrated Experiment Simulator Requirements:

DERIVE INTEGRATED EXPERIMENT SIMULATOR REQUIREMENTS

This section of the Report will contain requirements which the experiment may have for data or support from other experiments, special support equipment, or Space Station facilities. Examples of this could include specially formatted science data, pointing

data from Space Station Guidance and Control, or operational data from experiments or systems for synchronization of activities. There may be a requirement for a group of experiments to be activated and actuated simultaneously, or in sequence, which would necessitate the generation of outside controlling signals.

Since this support may not automatically be available in the PTC as it will be in the operational environment, it is important to specify the inputs which the simulator will need for proper operation. When the hands-on media Functional Specifications are reviewed and finalized, the integrated experiment requirements from all the simulators will be scanned to ensure that each Functional Specification contains the functions necessary to satisfy the integrated operational requirements of the other simulators.

PI-Provided Training Objectives:

LIST THE PI-PROVIDED TRAINING OBJECTIVES

In most cases, these should be available directly from the applicable data item in the Experiment Database. If not, the objectives could be obtained directly from the PI. The list should consist of all objectives, both those to be trained on simulators as well as academic objectives. If the PI is providing an experiment simulator, he or she should also provide the objectives which are considered appropriate and necessary to train with it. These initial PI-provided objectives will be an input to the Functional Specifications for each simulator. They will be considered along with objectives derived through Training Analysis in order to produce a cohesive and comprehensive set of training objectives for each simulator.

Experiment Data Outline:

PRODUCE A SUMMARY OUTLINE DESCRIPTION OF AVAILABLE EXPERIMENT INFORMATION.

This is a brief section which identifies experiment-specific data items available from the Experiment Database. The outline will serve as a guide for subsequent analysis efforts; showing where data is to be found, how much exists, is readily available, and by their absence, which data items must be obtained from the PI.

4.1.3 Experiment Overview Report Procedures Summary

Produce an Experiment Overview Report from data items in the Experiment Database. The EOR should address the following topics:

- Experiment overview
- Experiment development status
- Integrated experiment simulator requirements
- PI-provided training objectives
- Experiment information outline.

4.2 Simulator Approach Synthesis

Simulator Approach Synthesis is a process which examines the training requirements derived from front-end training analysis for each experiment, and integrates them with each other and with real-world constraints such as PTC policies, status of experiment development, cost-effectiveness strategies, and other external factors. The output of this integration, or synthesis is a preliminary approach for each simulator, documented in a Simulator Approach Document (SAD) for each simulator that will be used to train an experiment in a mission increment. This approach will be an input for the development of top-level simulator requirements and will serve as a generalized game plan for all requirements definition and related activities. As a side-product the synthesis process will produce a revised hands-on media Functional Specification for each simulator. In so doing, it will also unify all the training objectives for an experiment simulator into an integrated conceptual whole which can be communicated to JSC for inter-center training coordination. Figure 4-2 illustrates the synthesis process and its products.

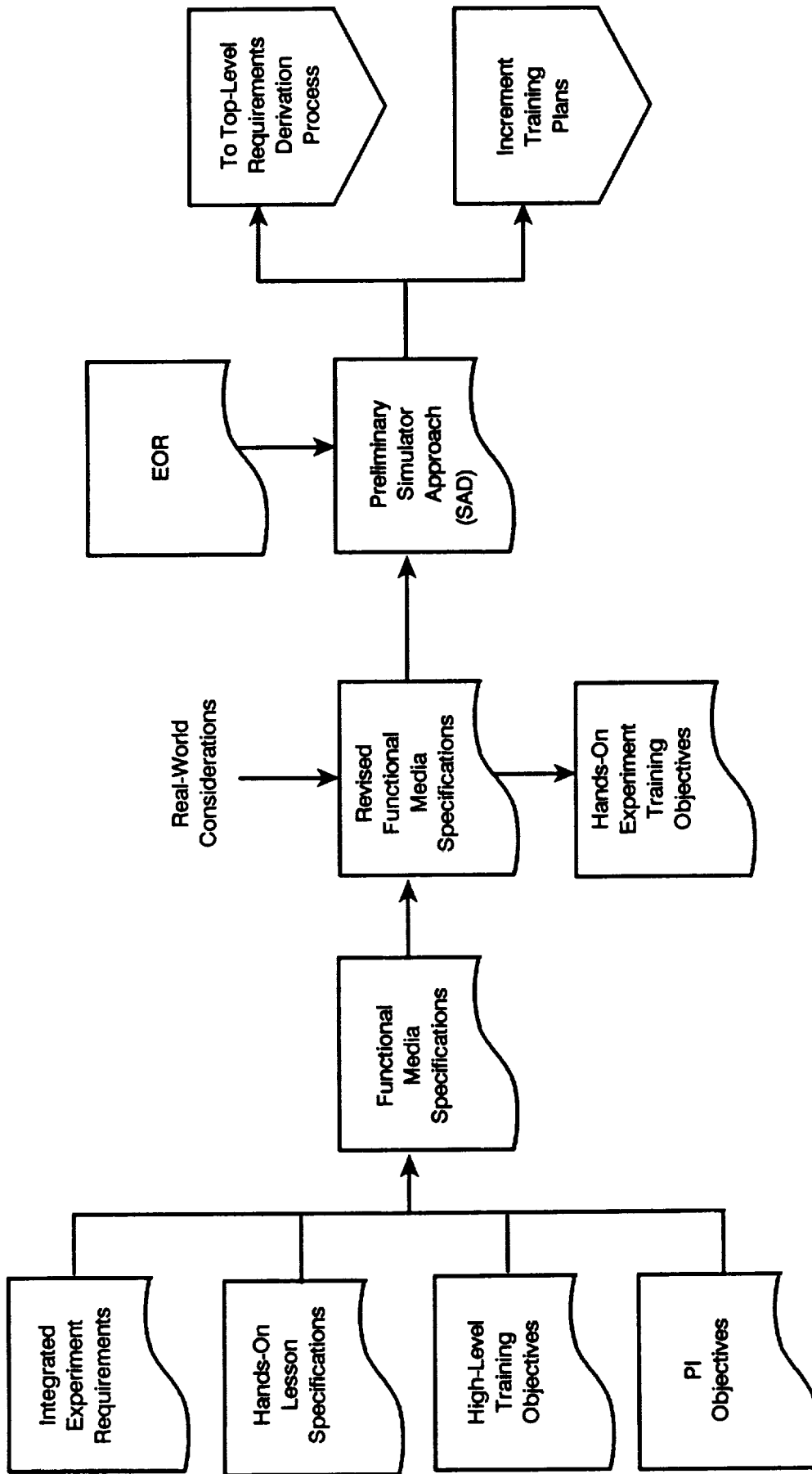


Figure 4-2. Development Flow for Simulator Approach Synthesis

The products of this process (and earlier ones) will also be useful in coordinating simulator development efforts between the PTC and the PIs. The EOR will flag significant training scope and design details of PI-developed simulators to PTC developers. The PI in turn will receive guidance to ensure that:

- a) the simulator will be supportable by standard PTC facilities
- b) the simulator will satisfy integrated simulator requirements
- c) the simulator's coverage of experiment training objectives will complement coverage supplied by the PTC.

This guidance will ideally be embodied in the form of the hands-on media Functional Specification for each simulator; listing all the simulator functional requirements necessary to satisfy the training objectives allocated to it. PTC interface requirements will be specified by an ICD (to be supplied by PTC programmatic sources). If the finalized Functional Specification is not available early enough to aid PI simulator development, its component parts can be supplied instead. These would consist of preliminary Functional Specifications, hands-on training objectives, and integrated simulator requirements from other-experiment EORs.

It should be noted that the simulator approach procedure only specifies the points in the methodology where various factors should be considered. It does not, however, specify the time that these inputs must be available in terms of the training development schedule. For example, if a Lesson Specification is not available at the time the simulator approach is first considered, then the specification may be considered at this place in the methodology when it is available, and reflected downstream to affect the final simulator requirements.

4.2.1 Hands-On Media Functional Specification Review

REVIEW AND REVISE THE FUNCTIONAL SPECIFICATIONS FOR EACH
EXPERIMENT SIMULATOR

The initial focus of the requirements synthesis effort is to produce a finalized Functional Specification for each simulator to be used to train a particular experiment. To do this, the Functional Specifications supplied as a product of Instructional Plan Development are modified by the following inputs and considerations:

Hands-On Lesson Specifications: These are descriptions of the hands-on training to be supplied for each experiment, produced during Syllabi Development. They include overall instructional strategies and methods to train each objective in a lesson. As such, they may indirectly levy requirements upon the simulator functionality. For example, a Lesson Specification could specify the implementation of a certain malfunction in the experiment processes in order to train for off-nominal conditions. In another instance, it might describe a certain instructor action based on certain data available to him or her. If the malfunction or the instructor data input is "new" to the Functional Specification, these capabilities may have to be added. In this regard it should be noted that revision is a two-way street. In other words, the analyst may decide to modify the Lesson Specification as a result of his or her analysis, rather than the simulator Functional Specification. In any case, the overall consideration is that the two specifications not conflict.

Integrated Experiment Simulator Requirements: These are the requirements for data or services between two or more experiments which were derived during composition of the EOR. They may not have shown up during the training analysis for each individual experiment since they involve requirements levied upon one experiment by another. One example of this could be a situation where data produced by one experiment simulator is needed by another. The supplying simulator must be required to calculate this data, and also provide it externally. Another case could be where the simulator must provide the information in a different format, or to a higher standard of precision. After scanning the EORs for all the experiments of an increment, the Functional Specification for each simulator is modified in order to accommodate the needs of the others in that increment.

"High Level" Training Objectives: These are objectives developed during training analysis which do not relate to only one experiment, but usually involve the simultaneous operation of multiple experiments. Examples include the development of teamwork skills, timeline skills, communications protocol, and resource balancing. While the primary mechanism for fulfilling these objectives will be lesson design, some extra functional capabilities may be required that have not been indicated through the objectives for individual experiments. These factors should be integrated at this point.

PI-Provided Objectives: These are objectives deemed by the PI for each experiment to be of primary importance to its operation or maintenance (they should be listed in a section of the EOR). Ideally, these requirements were incorporated during the early stages of training analysis, but they may have been unavailable at that time, or may have been changed or supplemented. For these eventualities, the PI-provided objectives not already integrated should be considered, and modifications to the Functional Specification made at this point.

Functional Specifications: In addition to all the external factors mentioned above, the collection of Functional Specifications for all the simulators dedicated to one experiment must be evaluated for characteristics such as internal consistency and scope:

Internal consistency: The hands-on media Functional Specifications should contain the necessary simulator functional and fidelity requirements to enable the training of a specific group of objectives.

These objectives will be associated on the basis of a common implied stage of training, and on similarities between their fidelity requirement levels. Even so, the separate requirements may levy differing levels of fidelity onto the same simulation component.

In that case, the training developer must select the degree of fidelity which will satisfy relevant training objectives in the most cost-efficient manner. As an example, a multi-function dial might have separate requirements for each of its functions.

If these requirements were to call for a static representation of every dial function except for one, the developer would have to decide how to best satisfy the training objectives. He or she could

- 1) represent the entire instrument as a static placard, off-loading the high-fidelity objective to another simulator

- 2) represent the entire instrument actively, thus meeting one requirement and exceeding the others
- 3) represent the instrument statically for all functions but one.

The final decision must be made against the overall situational background for the experiment and its increment. Rationales for all deviations should be recorded for future reference.

Scope: As mentioned above in "a", examination of the collection of simulator requirements in a Functional Specification may reveal the need to transfer objectives between the various Specifications dedicated to hands-on training for a particular experiment. Generally, the reason for such a move is to achieve a configuration for each simulator that makes more sense with respect to training sequence or simulator functionality. This was done for example, in the situation described above in "a," where the active instrument function was transferred along with the objective that required the capability, to another simulator which presumably has a more "active", computer-driven functionality.

In addition to manipulating individual objectives, a decision very likely to be made in the PTC environment would be to merge separate Functional Specifications to create a single simulator specification. Whereas during the training analysis process, it was appropriate to develop requirements in a very clear and academic fashion, during this stage it is necessary to consider real world constraints and conditions. Savings effected by designing one simulator rather than several may offset the inefficiency incurred by training students on simulators which are more capable than necessary.

The most likely situation would involve the need for a series of simulators with small, qualitative differences between them. In such a case, it would be easy to justify consolidating the specifications with lower fidelity and functionality into the ones requiring greater capability. This decision must be made by considering the relative differences between the various specifications, training complexity, current PTC training procedures, number of simulator copies required, the likelihood of frequent experiment changes, etc.

RECORD RATIONALES FOR ALL FUNCTIONAL SPECIFICATION REVISIONS. REPORT APPROPRIATE CHANGES TO SYLLABI DEVELOPMENT.

The resultant product of the above analysis should be relatively stable Functional Specifications for the simulators of each experiment. These revised specifications will define the scope of each simulator, and will provide a direct input to the development of top-level simulator requirements. At this stage, it is important to verify that a rationale for all changes to, or reconfigurations of, the Functional Specifications are recorded for future reference. Also, that any changes affecting the Lesson Specifications be transmitted to that activity. The Functional Specification review process is illustrated in Figure 4-3.

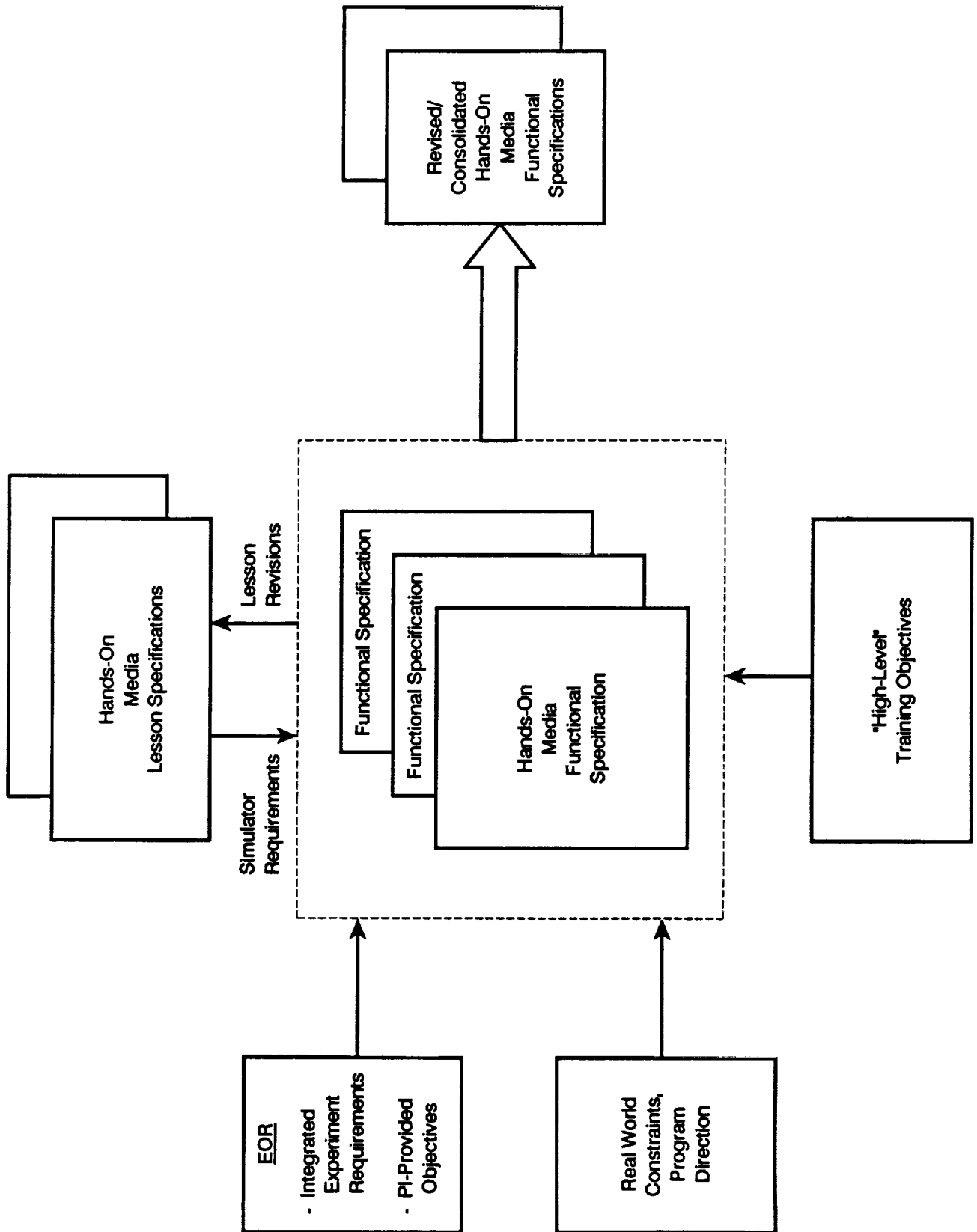


Figure 4-3. Hands-On Media Functional Specification Review Process

4.2.2 PTC to SSTF Coordination

PRODUCE A LIST OF TRAINING OBJECTIVES FOR EACH EXPERIMENT

Once a set of Functional Specifications has been approved for the simulators of an experiment, the training scope envisioned for those simulators as well as the scope of academic training will be transmitted to JSC. This scope description will consist of:

- a) A list of all academic objectives from the Training Database and
- b) A list of all hands-on objectives from the Training Database.

These two lists must be edited to reflect the changes made in the Functional Specifications. This includes the addition of any new PI-provided objectives and higher-level training objectives. JSC could then evaluate the training scope with respect to its own training plan. Return comments will be reviewed and any necessary changes will be input to the set of top-level simulator requirements, Functional Specifications, and/or Lesson Specifications.

4.2.3 Preliminary Simulator Approach

DETERMINE A GENERAL HARDWARE AND SOFTWARE APPROACH FOR EACH EXPERIMENT SIMULATOR

After all the functional requirements for a simulator have been finalized in a hands-on media Functional Specification, attention may be given to deriving a preliminary design strategy and hardware vs software allocation for the simulator which will satisfy its functional requirements. This initial plan will give an early heads-up for training resource allocation planning and provide a living framework of assumptions to support ongoing requirements development activities. The selected approach will be described in the Simulation Approach Document.

In order to clarify the following discussion, it will be helpful to refer to Figure 4-4, which illustrates a "typical" experiment configuration to be simulated in whole or part for training purposes. Shown is a Dedicated Experiment Processor (DEP) connected to various crew interfaces, and the instrument or assemblage of equipment necessary to perform the experiment.

**Flight & Ground
Crew Interfaces**

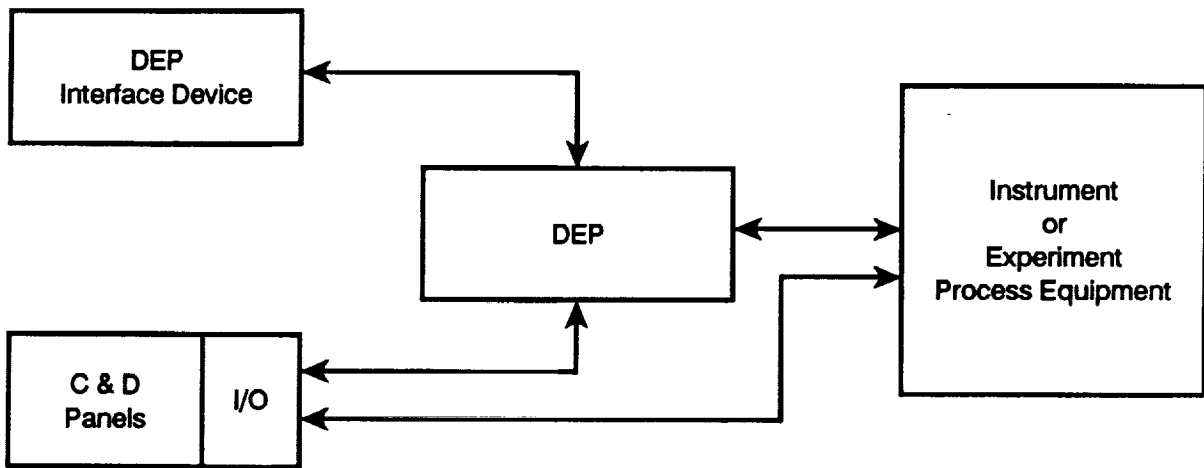


Figure 4-4. "Typical" Payload Experiment Configuration

Simulator Approach Selection must consider possibilities such as the use of flight hardware or software, total system simulations, and mixtures of the two. It should also account for the possibility of PI-provided simulators or parts thereof. The simulator Design and Development team should play a major role in this process so that Requirements Development will not be placed in the position of dictating design solutions. The overall goal should be to develop a simulator approach which provides the most cost-effective training solution, considering internal factors such as experiment type and development status, as well as external factors such as PTC resources and experiment equipment availability. While analysis is described here as a sequential process, it is actually iterative and interactive in nature. The various steps are inter-woven and will probably be accomplished in parallel. The major inputs to this decision making process include:

- Experiment Description and Status
- Functional Specification
- Dialogue with PI
- Available PTC Resources.

Using these inputs, simulator decisions may be made by considering factors which fall within four major categories:

(1) CONSIDER THE LEVEL OF PI INVOLVEMENT IN PTC TRAINING

Before any other consideration, it is important to ascertain what the PI is planning in terms of PTC training support for his or her experiment. This information should be available in the Experiment Overview Report which will discuss PI-provided training objectives, and whether a simulator or other training assistance will be supplied. It is important to verify that the PI-provided simulator both covers the necessary training objectives (as described in the Functional Specification) and conforms with PTC interfacing requirements. If discrepancies are noted, they should be reported to training management and/or compensatory measures should be planned by the training development activity.

(2) CONSIDER THE AVAILABILITY OF EXPERIMENT DEVELOPMENT
HARDWARE OR SOFTWARE

Even if a PI has no plans to develop simulators for PTC training of his or her experiment, he or she might still provide assistance in the form of experiment equipment. This could include an engineering simulator which was used for development, extra copies of equipment reproduced along with the actual flight equipment at Marshall's request, or equipment which is commercially available. The availability and utility of experiment development data such as flowcharts, listings, etc. should also be investigated since their availability may simplify approach decisions. This kind of information can be obtained from the EOR as well as from dialogue with the PI.

(3) CONSIDER EXPERIMENT DESIGN FACTORS THAT INDICATE THE SUITABILITY OF EXPERIMENT COMPONENTS FOR TRAINING (SIMULATION VS STIMULATION)

Once the PI resources available for training have been identified (above), their suitability for use in training may be evaluated. In most cases, this boils down to a simulation vs stimulation decision, since any experiment components used would have to be stimulated in the same manner as in the operational environment. The most important aspect in assessing the training suitability of an experiment component is usually the ease with which it could be supported in a simulated environment. Other considerations include cost, physical characteristics such as bulk or fragility, and maintainability.

As an example, consider an experiment using a commercially available computer to provide an interface between crew inputs and the DEP. If this component operates in an autonomous or semi-autonomous manner with relatively simple interfaces, it would be desirable for use in training, since directly updatable flight software could be loaded into it, just as in the actual experiment. Use of experiment DEPs for training can also avoid trainer concurrency problems in the software area; however, the DEP interface design must be considered. A DEP comprised of commercially available components and having simple or well-defined interfaces to other components could probably be used. On the other hand, if it was comprised of "home-grown" or modified components, maintainability could become an issue, and if its interfaces to other components were complex, use of the actual DEP might place heavy demands on simulation of the linking component. Flight equipment might be used for the linking component, but this link could well be a complex and expensive telescope or other sensor, unsuitable for training.

The experiment type often indicates the most advantageous approach. Materials Processing or Life Science experiments often use experiment hardware (such as furnaces, centrifuges, etc) with which the crew will directly interact, yet which may not have elaborate interfaces with the rest of the experiment hardware. In those cases, physical fidelity is of prime concern and actual hardware may provide this fidelity most easily. Care should be taken however, to determine the physical support requirements of the experiment hardware as well. Analysis may reveal requirements for fluids, vacuum, or zero-g which the PTC cannot support.

Use of flight software for training might ease development burdens and assure a certain degree of simulator authenticity, but it must be maintainable. If experiment software changes cannot be easily incorporated in the simulation, its presence could become a liability rather than an asset.

(4) CONSIDER HARDWARE VS SOFTWARE SIMULATION ISSUES

Leaving questions of the use of experiment resources aside, decisions must still be made as to whether a function is to be provided in hardware or software. In general, conditions which encourage the use of hardware for simulation include design stability. A well-established baseline for the experiment design would tend to encourage a hardware oriented approach, while the probability of numerous late design changes would tend to favor software solutions. Also, if there were high fidelity cue requirements, and the cue could be supplied by a physical representation, a simple physical simulation could be the simplest alternative. Software solutions on the other hand, are encouraged by requirements for versatile operation, such as the capacity to simulate malfunctions. In these situations, selected functions may be allocated to software simulation even if an overall hardware approach is adopted.

VALIDATE FINAL SIMULATOR APPROACH WITH MEDIA FUNCTIONAL SPECIFICATION. RECORD RATIONALE FOR DECISIONS.

As a final check, the selected approach must be compared with the requirements for that simulator levied by its Functional Specification. The simulator approach must not preclude any functionality demanded by the specification. Rationale for each approach decision should also be entered at this time. After

these final steps, the preliminary approach may be used as the structural model for the top-level simulator requirements document. Its hardware vs software allocations will also be used, together with the other simulator designs for an increment, in PTC resource utilization planning. This consideration encompasses the allocation of development as well as operational training resources.

The Simulator Approach Document should be organized as follows:

- a) Introduction
- b) Experiment Overview: operational objectives, major components, etc. (abbreviated from EOR)
- c) Basic Approach: brief overview of simulator, describing the general approach to simulator development, including hardware versus software allocations, and actual versus simulated equipment allocations
- d) Simulator Element Definition: description of all major simulator components (DEP model, C&D panels, Instrument Model, etc.), their purpose, general fidelity, and the organizations responsible for their development
- e) Design Approach Rationale.

4.2.4 Increment Training Plan

PROVIDE INPUTS FOR INCREMENT TRAINING PLANS

At a certain point in training development for an increment, enough is known about how training is to be conducted to allow plans to be made for PTC use. Planning includes scheduling resources for the development of courseware, lessons, simulator software and hardware, and instructional materials. Plans must also project requirements for trainers, classrooms, and utilities. Training tracks and learning sequences must be coordinated within PTC limitations. The information needed to perform this increment planning will come from the Lesson Outlines or Specifications, hands-on media Functional Specifications, and the basic approach selected for each experiment simulator including PI-provided simulators. Any problems arising from resource constraints or PTC programmatic limitations will be resolved through corrective feedback to the plan inputs. Figure 4-5 illustrates this process.

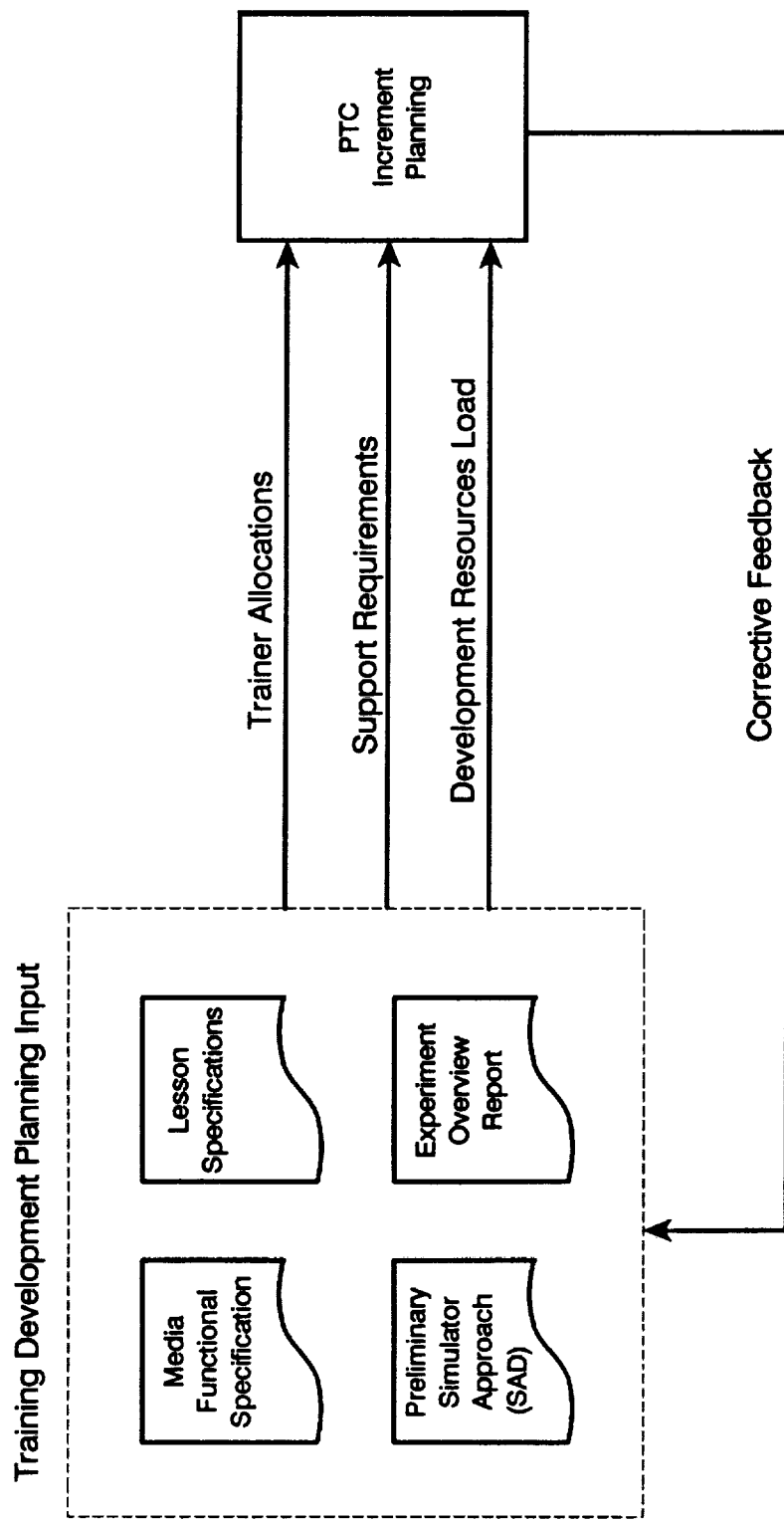


Figure 4-5. PTC Increment Planning from Training Development Input

A primary input to increment planning, Lesson Specifications detail what is to be taught for an increment, and the media with which it will be taught. This can include instructional aids such as flip-charts, exhibits, and slides, as well as general media categories such as classroom, CBT, or simulator. The sum of Lesson Specifications for an increment will be used to project the amount of classroom and CBT resources required; and courseware and instructional materials which must be developed.

The hands-on media Functional Specifications, taken together define the numbers and characteristics of the total simulators to be hosted in the PTC during an increment. This, along with the basic approach planned for each simulator, and information from the EOR on PI-provided simulators, can be used to define simulator support requirements and anticipated loads on simulator design and development resources.

DEVELOP PRELIMINARY PTC TRAINER ALLOCATIONS AS AN INPUT
TO INCREMENT TRAINING PLANS

Preliminary allocations of trainer resources (i.e. PTC trainers) can be made for the various simulators to provide a "first cut" trainer utilization plan. While the individual experiment simulators can be deployed within the PTC with a great degree of flexibility, they will generally be placed into trainers corresponding to a pre-defined level of training and training mission. Figure 4-6 shows the relationships between representative levels of training, training media, and the PTC trainer architecture. The relationships depicted are not hard and fast, but reflect general assumptions about the roles of different kinds of training media and about the roles of the various PTC architectural components.

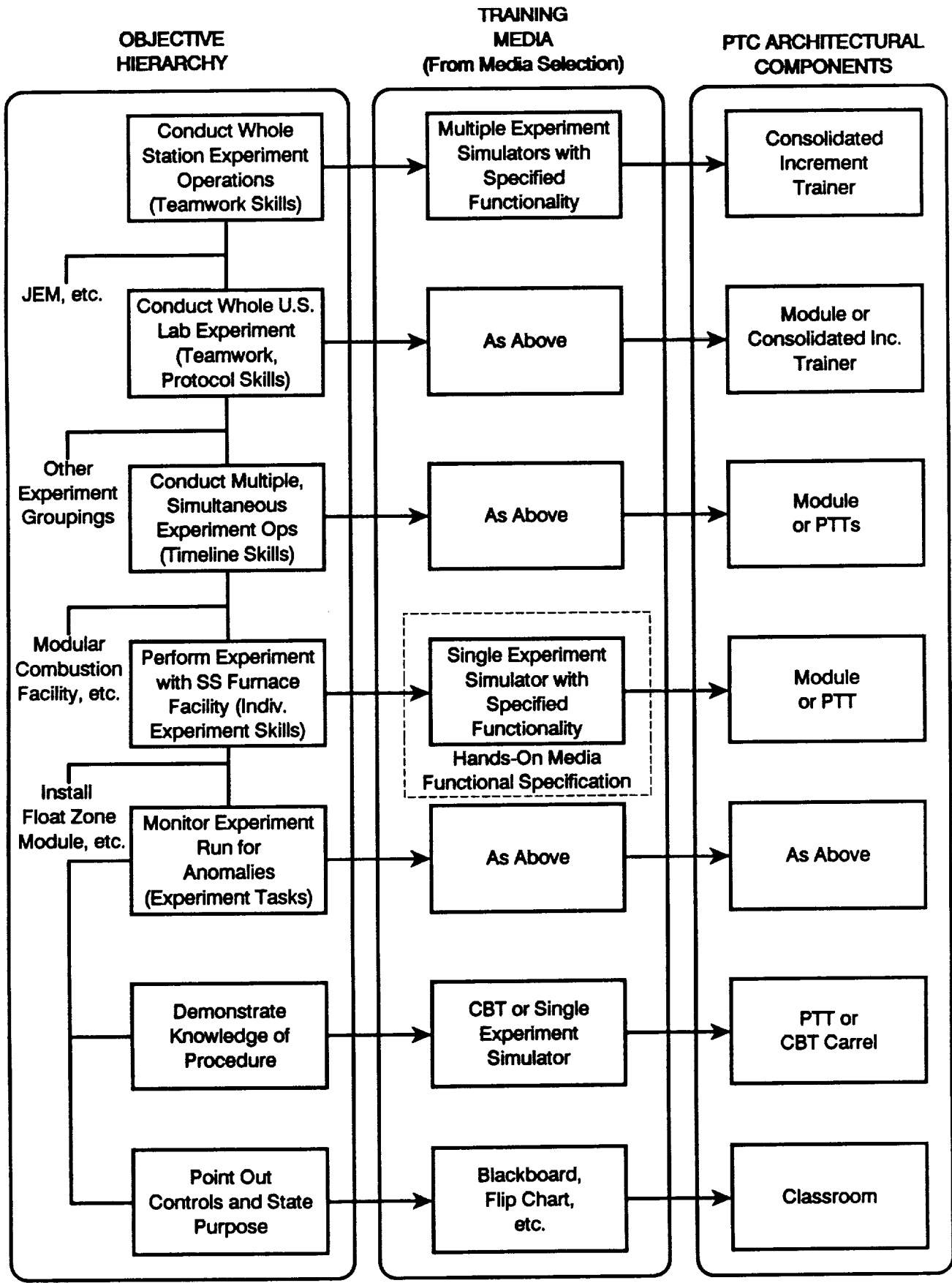


Figure 4-6. Relations Between All Levels of Objectives, Training Media, and PTC Trainers

Figure 4-6 implies that the requisite fidelity and functionality of the experiment simulators will be influenced by the requirements for training the different levels of learning objectives. These simulators may then be allocated to various PTC trainers according to the dictates of resource planning, and the roles for which the trainers have been designed:

Consolidated Increment Trainer: This refers to a configuration of interactive trainers representing the US, ESA, and JEM Labs. It can support a full complement of experiment simulators for a single increment. This configuration will typically be used to train skills necessary to operate the entire payload complement according to a given mission timeline. This will involve resource juggling, coordinating experiments' operations, and complex interactions between the station, and various ground facilities. Of course, lower level objectives can also be accommodated, and probably will be for maximum resource benefit. This configuration will also be used to validate integrated Space Station procedures for experiment operations.

Module Trainers: These are independent US, ESA, and JEM Labs trainers. Each trainer can support a full complement of experiment simulators for a single increment. These trainers will be capable of timeline validation, training coordination and communication type skills involving multiple experiments, but not training related to issues concerning the Space Station as a whole. It is anticipated that they will be extensively utilized for single-experiment operations training as well. These trainers will typically be the facilities used to train tasks with the highest fidelity cuing requirements.

Part-Task Trainers (PTTs): These are standardized devices, each capable of supporting one or two payload racks and a console or workstation to control the simulation of individual payloads. As such, they will be able to train for situations involving a limited number of simultaneous experiments. It is anticipated that they will find extensive use providing initial operations training for later flight increments, familiarization and procedures training, and refresher training for payloads experiencing last-minute changes.

CBT Trainers: Computers running instructional software ("courseware") which may drive audio and video courseware from an optical disk. They are used for training academic instructional objectives (i.e. objectives not requiring hands-on involvement). CBTs are typically implemented with desktop computers.

Attached Payload Trainer: This is a support environment for simulators of payloads mounted to the Space Station outside of the Labs. It will have minimal flight crew interface due to its primary mission of supporting ground controller training. Since

the example objective hierarchy illustrated is oriented toward flight crew training, it is not shown in Figure 4-6. While there is little ambiguity about the simulators which this trainer will house, it should be noted that if it is not available for a given experiment simulator, it may be possible to house the simulator in one of the other trainers.

POIC Trainers: Seven console trainers are planned for the PTC to support ground controller training for payload-specific operations.

As a general rule-of-thumb, it can be assumed that a typical trainee will encounter the following organization of curricula:

- a) General science and familiarization training in the classroom and on Computer-Based Trainers.
- b) Procedural and refresher training on Computer-Based Trainers.
- c) Initial experiment operations, nominal and contingency operations on individual experiments or very small groups of interactive experiments; with the PTT. Communications with ground controllers as necessary.
- d) Nominal and off-nominal experiment operations on the Module Trainers. Mission timeline training, communications protocol, teamwork skills between crew members.
- e) Full Space Station payload operations training on the Consolidated Increment Trainers. Contingency training for system malfunctions, payload malfunctions, Mission timeline training. Coordination between experiments in separate labs and ground facilities for resource sharing, data transfer. Communication with and between ground facilities.

For maximum flexibility in resource allocation, the major training elements (PTTs, Module, Consolidated) will be designed with similar I/O facilities and support capabilities. The trainers will supply electrical power and rudimentary pneumatics as required, but there will be no plumbing for fluids provided. A simulator designed to work in one trainer, however, should be capable of operating in the others.

4.2.5 Simulator Approach Synthesis Procedures Summary

a) Revise hands-on media Functional Specifications to incorporate:

- Integrated experiment simulator requirements
- Lesson Specifications
- High level training objectives
- PI-provided objectives.

b) Compile hands-on and academic objectives for coordination with JSC SSF training program

c) Develop a general hardware and software simulator approach considering factors such as:

- PI contributions to experiment training program
- Experiment hardware and software availability
- Experiment design
- Hardware versus software issues.

e) Develop inputs to Increment Training Plans:

- Preliminary simulator allocations to PTC trainers
- Development resources requirements
- Simulator support requirements.

4.3 Simulator Top-Level Requirements Document

The Simulator Top-Level Requirements Document (STLRD) defines the overall methodology of each experiment simulator. It does this by tying together information set forth in the Simulator Approach Document (SAD), the Experiment Overview Report (EOR), and the Functional Specification. The SAD will supply the simulator skeleton, its major components and the strategy for their development. The Functional Specification will supply the functional simulator requirements to be allocated to the various simulator components defined by the SAD. Lastly, the EOR will provide a general experiment description, including data on experiment interfaces, both internal and external which will be used to determine the required inputs and outputs for the various simulator functions. It is not intended that this document requires a great deal of original effort, but rather that it be created largely by integration of the analytic products mentioned above (see Figure 4-7). The major analytic responsibility in assembling this document is to map the requirements from the Functional Specification onto the appropriate simulator components.

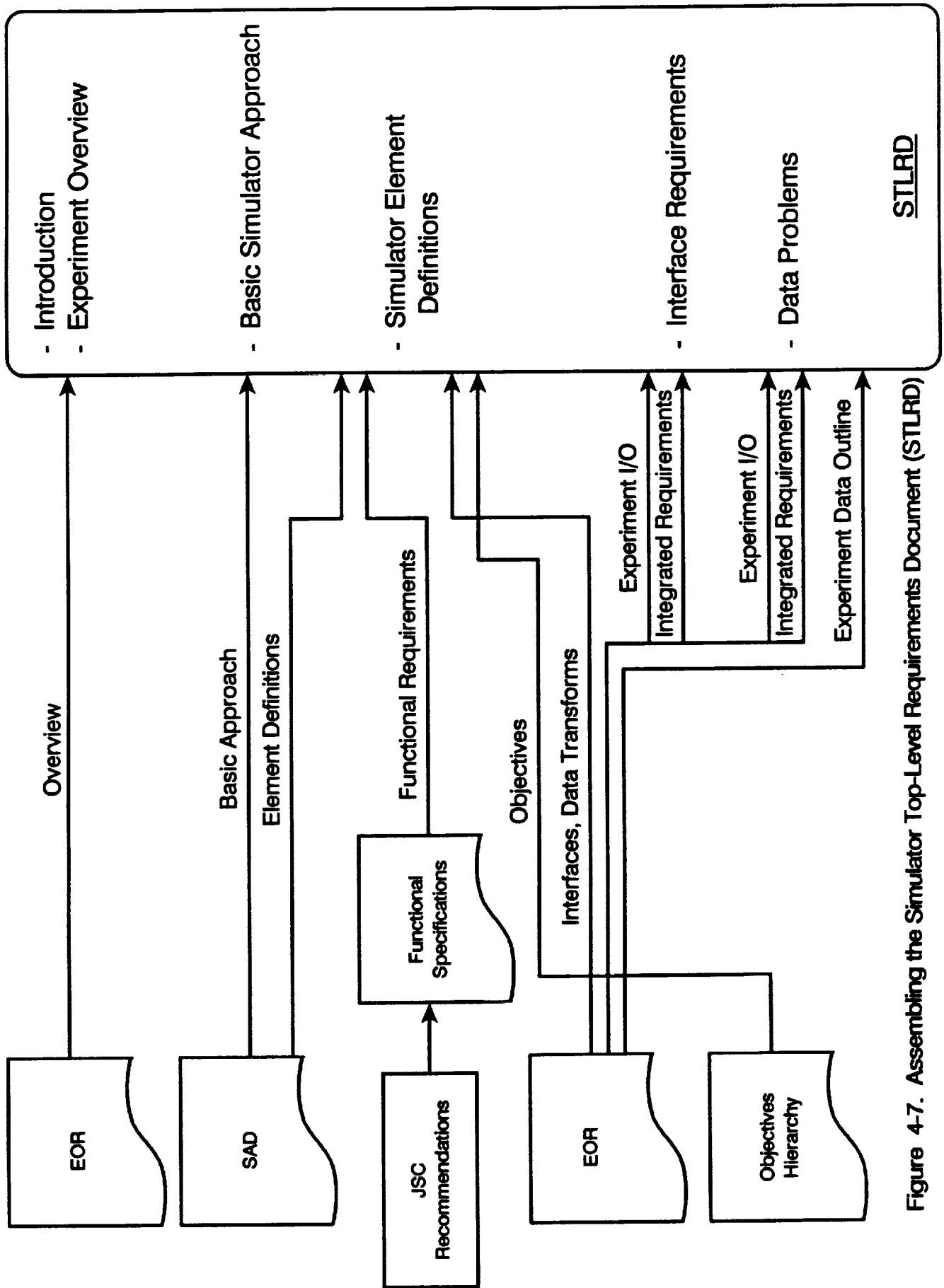


Figure 4-7. Assembling the Simulator Top-Level Requirements Document (STLRD)

The STLRD will be organized as follows:

SECTION 1 - Introduction

- purpose
- scope
- applicable documents.

SECTION 2 - Experiment Overview

- flight hardware and software components
- crew interfaces
- experiment functional objectives.

SECTION 3 - Basic Simulator Approach

SECTION 4 - Simulator Element Definitions

- element descriptions
- organizations responsible for development
- description of simulator element sub-functions
- tables relating simulator functions to Training Objectives and Functional Requirements.

SECTION 5 - Interface Requirements

- internal interfaces
- external interfaces.

SECTION 6 - Data Problems

4.3.1 Document Assembly

ASSEMBLE EXPERIMENT OVERVIEW AND BASIC SIMULATOR APPROACH

The Experiment Overview section of the STLRD can be constructed directly from applicable portions of the Experiment Overview section of the EOR. The purpose of the Experiment Overview is simply to summarize in condensed form, the nature of the experiment to be simulated and its general configuration. The EOR text ideally should be designed to allow its direct inclusion, though some editing may be required.

The Basic Simulator Approach can likewise be constructed directly from the Basic Approach section of the SAD. As with the EOR, the text should be designed to facilitate its direct inclusion.

ASSEMBLE SIMULATOR ELEMENT DESCRIPTIONS
MAP FUNCTIONAL REQUIREMENTS TO THE APPROPRIATE SIMULATOR
ELEMENT

The simulator element definitions are the heart of the STLRD. They consist of a top level description of each major simulator component and their sub-functions. The simulator element sub-functions state the general methods to be used to satisfy the training objectives and requirements with which they will later be correlated.

The top level element descriptions can be lifted directly from the Simulator Element Definitions in the SAD. These descriptions include the element purpose, general fidelity, and the organizations responsible for their development. Next, the requirements in the Functional Specification for this simulator are allocated to the appropriate simulator element(s), and sub-functions based on them are written for each element. In other words, each major simulator element is broken down into components which are largely defined by the functional requirements which they must meet. The EOR will prove useful in this effort as well, providing data on interfaces, data transformations, and references to detailed experiment data.

At this point, inputs from JSC should also be considered in terms of their effect on the training objectives and thus, the simulator requirements. After the Functional Specification for each simulator is finalized during Simulation Approach Synthesis, a list of training objectives is sent to JSC for coordination with the SSTF. Any problems with the scope or nature of the objectives should be reported back to the PTC so that program modification may be considered. This input should ideally be available in time for its inclusion into the STLRD. If a change is decided upon, it should be reflected in the Functional Specification for that simulator, and then, in the appropriate simulator element functions. When complete, the element sub-functions in the STLRD should reflect the modified Functional Specification, stating how each experiment capability will be simulated, and the fidelity of all outputs.

At the end of the Element Definition section, the functional requirements from the simulator Functional Specification and their parent Training Objectives from the Objectives Hierarchy will be correlated in a series of tables with the element function or functions in the STLRD which satisfy them. This will

serve as a guide for Verification traceability, and also will provide a reference to the parent requirements, when the element sub-functions are broken down into detailed simulator requirements in the ESRD.

ASSEMBLE SIMULATOR INTERFACE REQUIREMENTS

This section is a textual description of the interfaces between the simulator elements, and between the simulator and other simulators, hardware elements, and the PTC system. The purpose of this section is to provide a detailed description of I/O relationships so that later analysis can result in specific simulator requirements. Information for this section will come from the EOR, which describes experiment I/O, and integrated experiment simulator requirements, as well as from established policies and architecture of the PTC.

IDENTIFY PROBLEMS IN THE AVAILABLE EXPERIMENT DATA WITH RESPECT TO DERIVING ADEQUATE SIMULATOR REQUIREMENTS

This section will be used to record instances where the available experiment data is deemed insufficient for establishment of top-level and/or detailed simulator requirements. A survey to identify these insufficiencies can be conducted by inspection of the Experiment Data Outline in the EOR, which describes the experiment data available, and the explicit I/O and data transformation data in the EOR Experiment Overview. This information can be compared with the simulator elements and their sub-functions described earlier in the STLRD. Obviously, if not enough is known about an experiment feature to allow even a top-treatment of its simulation, then a data deficiency exists. Beyond this, if the data is sufficient to allow top-level coverage, there still might not be enough to outline its detailed implementation in the ESRD. Even if a proper Task Analysis has been made, the experiment information required to develop the tasks may not be sufficient for detailed implementation of experiment functions.

For hardware items and functions, required data could include drawings, schematics and associated parts lists; functional descriptions of equipment operation, support requirements, and detailed I/O lists including mnemonics descriptions with typical, minimum, and maximum values. For software features, required data

could consist of specific data transformations, I/O lists, iteration rates, data modes, display screens, flowcharts and source code (if available), interface commands, flags, data sets, scene control parameters, and explanations of software functionality.

The key to a proper assessment of experiment information is to analyze simulator data needs on a functional basis, determining what must be known about the experiment to implement each function in the manner prescribed by the STLRD and the Simulator Approach Document. Allow the required simulator functionality to define the type of information, and the level of detail required. In that way, the search for data can be restricted to that which is actually needed to implement the simulator.

Data problems should be identified in this section of the STLRD by:

- a) briefly describing the experiment function or component for which data is lacking
- b) relating the experiment function or component to the method or candidate method of implementation described in the Simulator Element Definitions
- c) describing the data needed in terms of the function or capability to be simulated.

4.3.2 PI Interview

STRUCTURE AN INTERVIEW WITH THE PI TO CORRECT ANY DATA INSUFFICIENCIES

Once data insufficiencies have been identified in the STLRD, they should be brought up to the PI in an interview. Ideally, this interview will yield all the information which might be needed to complete the STLRD and write the ESRD. However, even when data problems are well defined, it may be difficult to elicit all the information needed in one "go-round" with the PI. This could be because the PI omits subtle but important details in his or her answers, or fail to mention constraints, conditions, extenuating circumstances, etc. In addition, the interviewer may lack sufficient experiment understanding to ask all the necessary questions. While there is no way to guarantee that an interview will yield 100% of the necessary information, adherence to the guidelines suggested below should improve the chances of success:

Structure the interview questions around the experiment and/or simulator functions which have been identified as problem areas in the STLRD. Concentrate on the type of information and the level of detail specified in the STLRD.

Ask questions in a top-down sequence, addressing the major simulator elements (such as the DEP), and the more general and conceptual questions first, before proceeding to more specific questions about an element's sub-functions. This will allow both the interviewer and interviewee to converge on a common mindset before detailed questions are asked.

Explain why you need certain information in terms of the method of simulation so that the PI will better understand your data needs. Many times, if the PI understands where you are trying to go with certain questions, he or she can more easily give you the information needed. In addition, if the PI understands the basic simulator approach, there is a greater possibility that he or she will understand the effects which later experiment changes could have on the training program and advise training personnel accordingly.

Ask the PI about the probability of later changes to the experiment, especially in the areas covered during the interview. In addition, ask the PI to keep you up-to-date concerning any experiment changes, especially those relating to the functions discussed in the interview.

Avoid asking too many questions which can be answered with a yes or no. Often, this kind of answer represents an over-simplification by the expert, masking important details of a situation, or omitting qualifiers, contingencies or alternate possibilities. Also, it represents a minimum information return. Since the purpose of the interview is to elicit as much pertinent information as possible, the interviewer should design questions which require complete responses that explore all sides of an issue.

In addition to the above guidelines, the following are some generic questions which can help to "round out" an inquiry:

- a) What are the major elements to consider when performing a task which utilizes the experiment function under discussion?
- b) What are the interrelationships or dependencies between the task elements?
- c) What are the stimulus cues used by the trainee when interacting with the experiment function?

- d) What outputs can be expected from the experiment function?
- e) What constrains the use of the experiment function?
- f) What conditions must be accounted for when using the experiment function?
- g) What function-consequence relationships exist, that is, if I use this function, what will result?

4.4 Experiment Simulator Requirements Document

DERIVE DETAILED SIMULATOR IMPLEMENTATION REQUIREMENTS AND DOCUMENT THEM IN AN ESRD

At this point in the simulator development process, the major part of the analysis effort has been completed. The basic simulator approach has been determined and its various elements defined. Ideally, all experiment data necessary for simulator development has been identified and collected. The final step is to use this information to develop hardware and software implementation requirements in sufficient detail to allow simulator design and development efforts to proceed.

The ESRD organizes these requirements under the same simulator elements and sub-functions defined in the STLRD. Since the general simulation method for each sub-function of each element has been previously determined, all that is needed are descriptions of the specific requirements to accomplish each function. For software models, this consists of whatever is necessary to define its inputs, outputs, and behavior. For hardware components, this will mean system schematics, mechanical drawings, parts lists, and any other information about the actual experiment needed by D&D to create simulator hardware specifications.

DERIVE DETAILED REQUIREMENTS FOR EACH SIMULATOR ELEMENT

Analysis will be conducted by considering each simulator element and its subfunctions in turn; supplying the detailed information necessary for its realization in hardware or software. The tables compiled at the end of each Element Definition in the STLRD can be used to trace top level simulator functions back to their

parent requirements in the Functional Specification. These requirements will provide detailed information on required cues, cue fidelities, display formats etc. which will help to derive the final, detailed simulator requirements.

The heart of the ESRD will be the implementation requirements for each of the Simulator Elements. Since an element could consist of software or hardware, and simulated or actual flight equipment, the format for the implementation requirements will differ for each element. However, most elements will contain some or all of the following:

Interface Requirements: These are the detailed input and output requirements necessary to satisfy the simulator fidelity and functional requirements spelled out in the STLRD, and supported by the Functional Specification. This category includes interfaces internal to the simulator, such as between software modules or between a software module and simulator hardware. Also included are external interfaces to PTC support equipment or other simulations. Interface requirements typically consist of I/O Lists specifying mnemonic, range, resolution, units, description, and destination. Diagrams and textual explanations may also be included.

Modeling Requirements: These are the detailed command input and parameter output relationships necessary to fulfill the simulator fidelity and functional requirements. They define the data transformations and control structures which comprise the bulk of the experiment simulation. Based primarily on experiment transformations and structures, they include the required functionality for malfunctions, as well as control and contingency modes of the simulator. While this category of requirement can be represented in many forms, it is usually expressed as a textual explanation of functions, inputs, and outputs, supported by mathematical equations, Boolean equations, and truth tables. Complex functions may be represented by conceptual flowcharts, or experiment information such as flowcharts and data tables may be available from the PI and directly applicable.

Hardware Mockup Requirements: These are representations of experiment hardware in sufficient detail to allow simulator Design and Development personnel to derive a specification for manufacture. These requirements may include panel drawings, system schematics, parts lists, mechanical drawings, signal input and output lists, and textual explanations of required fidelity levels and hardware functionality. This section should also include any required support equipment for training.

ESRDs will be written for a wide variety of experiment simulators and implementations. Since the purpose of the ESRD is to provide

enough information in the most convenient format for simulator development, not all ESRDs will be structured in the same way. The structure of the ESRD should be sensitive to the nature of the experiment and to the characteristics of the information available; thus its format should not be rigidly constrained. The following therefore represents a generalized description of a prototypical ESRD rather than a strict template to follow:

SECTION 1 - Introduction

SECTION 2 - Simulator Elements

- **Software Simulations**
 - Instrument Models
 - DEP Model.
- **Stimulated Experiment Hardware**
 - Crew Interface Module
 - DEP
 - Experiment Instrument.
- **Hardware Mockups**
 - C&D Panels
 - Process Machinery.
- **Simulator-unique Software**
 - Scene Generation.

SECTION 3 - Design Considerations

SECTION 4 - Appendices (extensive data items)

- Data Tables
- Flowcharts.

4.5 Simulator Top-Level and Detailed Requirements Derivation Procedures Summary

Assemble a Simulator Top-Level Requirements Document

- a) Integrate elements of the Experiment Overview Report, Simulator Approach Document, Functional Specifications, and Training Objectives Hierarchy
- b) Develop simulator element descriptions from elements described in the SAD

- c) Map simulator functional requirements to the appropriate simulator element
- d) Identify deficiencies in available experiment data
- e) Structure a PI interview to correct deficiencies.

Derive detailed requirements for each simulator element and assemble an Experiment Simulator Requirements Document

- a) Interface requirements
- b) Modeling requirements
- c) Hardware Mockup Requirements.

5.0 TRAINING DEVELOPMENT VERIFICATION METHODOLOGY

5.1 Introduction

The PTC Training Development Verification Methodology defines a process to verify that the PTC-hosted training requirements are being properly implemented during development. The purpose of verification, as part of the Training Requirements Development System (TRDS), is to provide NASA with systematic assurance that developed payload trainers will fulfill their role for PTC training in a correct, effective, and economical fashion. Verification is performed by a verification group that is detached from the development group. This verification group, known as the Verifier, provides NASA with an objective and independent perspective to assess the technical adequacy of the delivered products.

The verification process involves a series of activities interface with the development process itself, and supports a more orderly and efficient implementation because each development phase produces a verified baseline for the next phase. As shown in the TRDS Template described in the Program Concept, verification activities begin during the Training Requirements Analysis phase and end with the Simulator Acceptance Review (SAR). As a result of the verification activities, errors are typically uncovered early in the development cycle before they have a chance to propagate. This early discovery promotes improved reliability, greater visibility, and reduced life-cycle costs.

5.1.1 Verification Definition

This verification methodology is a customized methodology to fulfill the PTC training system development needs; and is based on current MSFC verification procedures as described in the PCTC Development Handbook [1], the SpaceLab Flight Software Test Plan [2], V&V industry standards as described in [3], and V&V guidelines as discussed in [4]. An important observation is that the term "verification" has different meanings and connotations within different organizations.

The term "Training Development Verification" for PTC is defined as:

- The process of determining whether or not the products of each phase of the development cycle fulfill the requirements established during the previous phase (as based on the IEEE Standard Glossary of Software Engineering Terminology) and

- The process of testing the simulator software to demonstrate that the software fulfills all requirements imposed by the requirements specification.

In contrast, Simulator Validation is defined as the process of evaluating the simulator to insure compliance with the training objectives and overall simulator requirements. Informally, these terms have been described as - "Are we building the product right?" (Verification) and "Are we building the right product?" (Validation).

5.1.2 Levels of Verification

The verification process is organized into three major levels of verification activities:

a) Increment-Independent Verification Planning: Prior to the development of the first SS increment training system, the verification process includes a one time activity to generate a Generic Master Verification and Test Plan. This generic plan will guide the verification process during the development of all the training systems, and will be a detailed expansion of this Verification Methodology. The generic plan would be updated periodically as required. The Verification Team will prepare a tailored Verification Test Plan for each SS increment training system. The Test Plan will describe any customized verification activities as required for that particular increment. During this time, the Verifier will also plan, procure, and develop desired verification tools for use within each Increment verification activity.

b) Specification Verification: The purpose of Specification Verification is to allow in-progress verification of the training development process. Specification Verification is an iterative process of determining whether the product of each development phase fulfills the requirements levied by the previous phase. The Verifier is interested in both the simulator and non-portion of training development. Specification Verification creates a series of verified baselines upon which the instructional products can be developed and tested, and provides NASA with the feedback they need to manage effectively. There are five stages of specification verification for each SS increment training system, as summarized below and described more fully in Section 5.2:

1) Training Objectives Verification: The purpose of verifying training objectives is to assess whether the objectives hierarchy for each experiment, as prepared

by the responsible PI, are a fair representation of the training needs for that experiment.

2) Instructional Plan Verification: The purpose of Instructional Plan verification is to determine whether the instructional media, with an emphasis on the computer-applicable portion of the Instructional Plan, represents a clear and accurate description of the training needs.

3) Simulator Requirements Verification: The Verifier analyzes the Simulator Requirements Verification is used to ascertain that the data systems requirements (both hardware and software) reflect the needs expressed in the Instructional Plan.

4) Design Verification: During Design Verification, the Verifier analyzes the simulator designs to verify the software design for technical adequacy and that it satisfies the Simulator Requirements.

5) Code Verification: The purpose of Code Verification is to allow a "code walk-through" of the code listings to determine whether the actual code implements the described designs.

c) Verification Testing: The purpose of Verification Testing is to plan and conduct tests to verify that the implemented software fulfills the simulator requirements. This testing does not include the testing responsibilities the developer. Verification testing is concluded with the Simulation Acceptance Review. At that time, the validation activities are initiated to validate that the overall training system fulfills the overall training objectives. Verification Testing is fully described in Section 5.3.

5.1.3 Verification Options

At the option of NASA, the verification process can be performed by any or all of the following:

a) A semi-independent verification group provided by the developer contractor. The verification group would be independent of the developer group, but both groups report to the contractor's program manager.

b) A semi-independent verification subcontractor procured by the developer contractor. Independence is enhanced if NASA explicitly tasks the developer contractor to use a subcontractor and maintains an active rope in overseeing the subcontractor progress and status. This option is

established as the default choice pending a decision to the contrary.

c) An independent verification contractor. True independence is achieved but at significant, and probably unnecessary, cost.

d) NASA personnel. True independence is achieved, but adequate personnel may not be available.

5.1.4 Depth of Verification

How much on-going verification is necessary? In general, if errors are detected early, the overall life cycle cost of simulator development is reduced and reliability is increased. However, it is possible to spend more on upfront verification activities than would be saved in reduced overall development costs. This methodology defines the total verification process, and the depth of verification activities is to be determined a part of the Generic Master Verification and Test Plan.

Dependent upon the size of (and budget available to) the Verifier, the Verifier will then follow this methodology to the level of detail necessary for each SS Increment. Good management judgement must be used with each Increment to achieve a good balance to accomplish the proper level of specification verification. For example, a specification verification team of one person would be inexpensive and serve as a low-cost insurance policy to uncover some, but probably not all, errors as an on-going activity during the development process. At the other end of the spectrum, a large independent contractor verification team could cost more than the development team.

In order to perform an adequate level of verification, the Verifier cannot wait for the final version of a document to perform the verification. Thus, the overall training system development plan must allow for interim and informal delivery of partially completed documents to the Verifier. Then the completed version of a document must be made available to the Verifier prior to the formal Review. The Verifier then completes the product verification and presents his or her findings at the formal Review. The amount of time available for verification of each final document product must be established during the requirements definition phase, and is dependent upon the level of verification detail (and attendant costs) desired by NASA and the complexity of the product being developed. Similarly, as the complexity of the product grows, the developer's allotted time cannot be minimized.

5.2 Specification Verification

For each specification to be verified, the developer will generate the specification and make preliminary draft versions of the documentation available to the Verifier to allow on-going analysis. Prior to the formal review of that specification, the Verifier receives and examines the completed documents and prepares an Analysis Report. The Verifier will prepare for the formal review and present the results of the analysis. Issues, problems, and potential solutions are to be highlighted. After the review, the Verifier will generate formal Engineering Change Requests (ECRs) to describe those specific changes as dictated by the Review Board. The schedule and milestones for this specification verification process is defined in the Program Concept and TRDS Template. The purpose, responsibilities, deliverable, and activities associated with each verification activity are described below.

A number of techniques for specification verification are effective, and range from simple manual analyses to fully automated procedures [4]. The selection of the desired technique is dependent upon the complexity and criticality of the product being verified. The breadth and depth of the specification verification process is highly dependent upon the amount of time available to perform the verification.

Manual techniques include reading, manual cross-referencing, interviewing, checklists, manual models, and simple scenarios. Independent reading, in itself, is an inexpensive and effective technique to expose the document to a different perspective and point of view. Manual cross-referencing involves the construction of tables and diagrams to clarify interactions, and is particularly effective to analyze small- and medium-sized specifications for consistency. Interviews are helpful with minimum effort to expose misunderstandings and high-risk areas for further examinations. Checklists are excellent for uncovering omissions and incomplete specifications. Developing manual mathematical modes are helpful when performing feasibility assessments. The use of simple scenarios help to show if the simulator would work effectively during training.

Automated techniques for requirements verification include automated cross-referencing tools which are used to capture specification data in a data base which can then be scanned for completeness and consistency. Examples of such tools are the structured analysis tools, such as Power Tools by Iconix and TeamWork by Cadre, available with the SS Software Support Environment (SSE). Other automated techniques for requirements verification, such as Requirement Simulators, are probably inappropriately time-consuming for use in a training system verification. Finally, automated techniques for code

verification include Code Analyzers which scan the code to verify the code is built according to prescribed standards.

5.2.1 Training Objectives Verification

VERIFY THE TRAINING OBJECTIVES REPORT FOR
OVERALL INTEGRITY, REASONABLENESS, AND
COMPLETENESS

Purpose: Training Objective verification is performed to ensure that the Objectives Hierarchy and Task Hierarchy developed during Training Needs Assessment activity are a complete and accurate reflection for the training needs of each experiment.

Responsibilities and Deliverables: As described in the Training Needs Assessment Methodology, the Developer produces the Training Objectives report which includes a Task Hierarchy and Objectives Hierarchy. The Verifier is responsible to evaluate the set of training objectives for overall integrity and completeness. The PI is also responsible for reviewing the Training Objectives report for correctness as based on his or her understanding of the experiment purpose. The Verifier will combine both the PI's observations and his or her own findings in a Verification Analysis Report, and updates and corrections.

Methods: The Verifier will examine the Training Objectives report as it becomes available, and review the document for its overall integrity. The Verifier will use engineering judgment to determine whether the Task Hierarchy provides a reasonable breakdown of the required training tasks. The Verifier will evaluate the hierarchy of training objectives to confirm that each required entry for each objective is present and clearly stated. The required entries are: Objective Statement, Behavior, Conditions, Standards of Performance, and Measure of Training Effectiveness. In particular, the Verifier will ascertain that each stated requirement is expressed in measurable or observable terms so that the training personnel can specifically determine if the training objective has been achieved or not. The Verifier will confirm that each task in the Task Hierarchy is traced back to some originating data item in the Experiment Data Base, and is traced forward to one or more objectives in the Objectives Hierarchy (Figure 5-1).

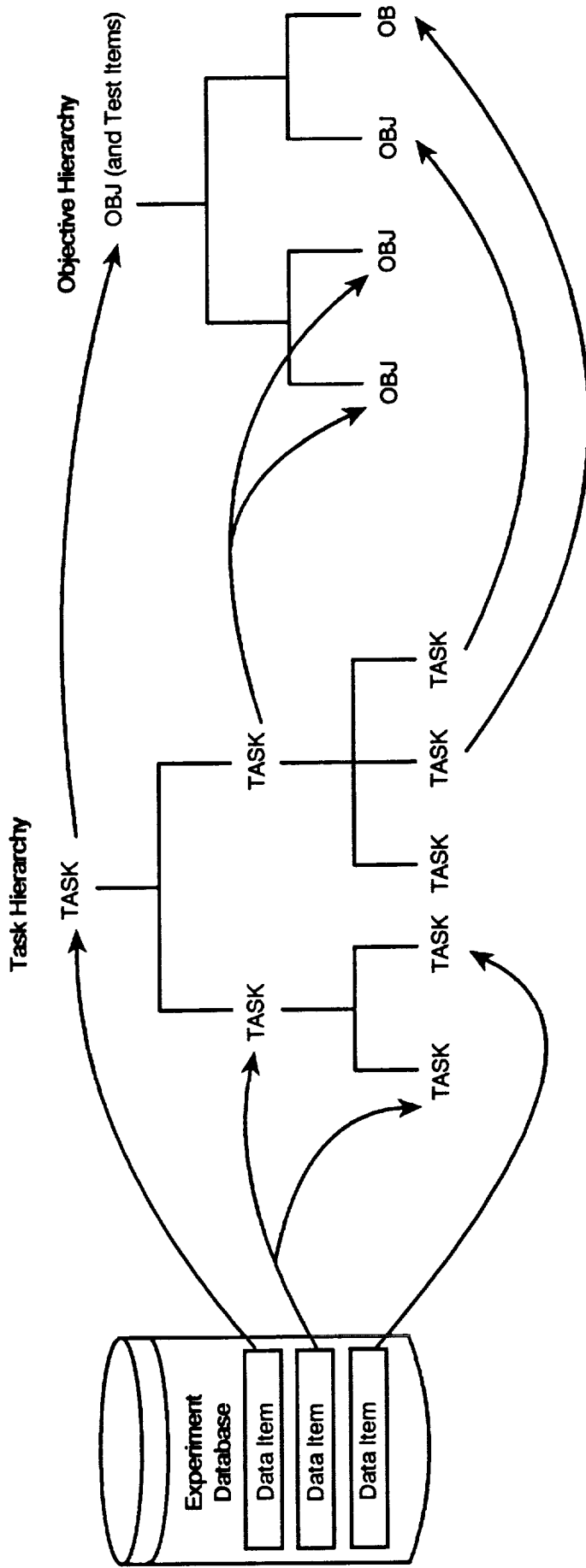


Figure 5-1. Training Needs Assessment Traceability

5.2.2 Instructional Plan Verification

VERIFY THE INSTRUCTIONAL PLAN REPORTS TO ASCERTAIN THAT THE SELECTED TRAINING TECHNIQUES AND INSTRUCTIONAL MEDIA ACHIEVE THE REQUIRED TRAINING OBJECTIVES

Purpose: Instructional Plan verification is the activity to ensure that the overall Instructional Plan, as described in various documents during the instructional planning process, provides a clear and accurate description of the selected training techniques and instructional media. The Verifier will examine the Instructional Plan with an emphasis on the computer-applicable portion of the plans to determine whether the plan will achieve the required training objectives. After this activity, the development team can develop simulator requirements with increased confidence in the accuracy and clarity of the Instructional Plan.

Responsibilities and Deliverables: As described in the Instructional Plan Development Methodology, the developer will produce three related reports: Instructional Methods and Media Specification, Lesson Specifications, and the Instructional Plan itself. The Verifier will review these documents to provide an independent perspective on the thoroughness, overall soundness, and balance of the plan, concepts, and approach. The Verifier is also responsible for reviewing this documentation with a focus on evaluating the computer-applicable portions of the plan in terms of risk and technical feasibility. The Verifier prepares the Analysis Report, and reports the analysis results at the Training Preliminary Requirements Review (PRR).

Methods: The Verifier will examine each report separately as it becomes available, and will review the document for its overall clarity. The Verifier will ensure that each document is complete and addresses all of the required information as described within the Instructional Plan Development Methodology. The Verifier will ensure that the plan includes techniques for determining if the instruction is effective. Where training results can be measured, the Verifier will ascertain that the recommended tests are specific, unambiguous, and quantitative whenever possible.

The Verifier will ascertain that the components of the planning documentation - academic media objectives, lesson specifications, functional hands-on media specifications - are traceable to the training objectives hierarchy (Figure 5-2). In particular, the traceability analyses will ascertain that:

- a) All academic objectives are allocated to one or more Lesson Specifications.

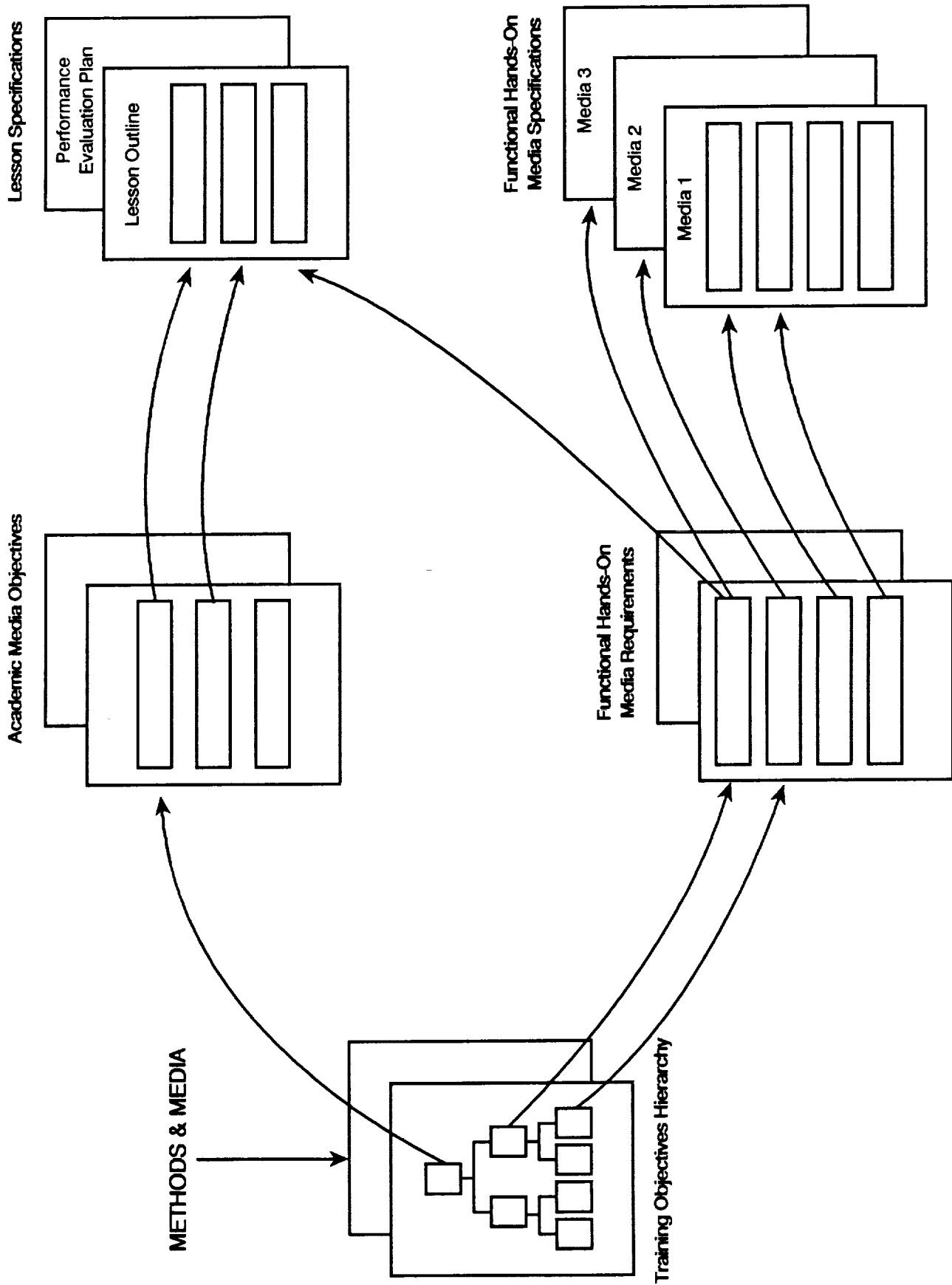


Figure 5-2. Instructional Plan Development Traceability

b) All hands-on objectives relating to the operation of a single experiment are allocated to both a functional Hands-On media specification, and one or more Lesson Specifications.

c) All hands-on objectives relating to operations involving more than one experiment are allocated to one or more Lesson Specifications.

The Verifier will examine the Instructional Plan to expose any potential requirements which are unjustifiably complex for development in the PTC. The Verifier can conduct trade studies to investigate alternative training concepts in terms of benefits, costs, and risks. Where possible, the Verifier will review those training requirements which address the user of flight equivalent equipment as opposed to the use of host-based software simulator to verify the technical approach is sound.

5.2.3 Simulator Requirements Verification

VERIFY THE SIMULATOR REQUIREMENTS DOCUMENTATION FOR TRACEABILITY, COMPLETENESS, CONSISTENCY, FEASIBILITY, AND TESTABILITY TO ENSURE THAT THE STATED REQUIREMENTS REFLECT THE INSTRUCTIONAL PLAN AND CAN BE USED TO PRODUCE A SOUND DESIGN.

Purpose: The Verifier conducts the Simulator Requirements Verification activity to ensure that the specified requirements:

- Reflect the needs of the Instructional Plan
- Can be used without ambiguity to produce a sound simulator design.

Since the simulator requirements is the critical gap between the training needs analysis and the simulator development activities, a special emphasis on the verification of the simulator requirements is desired.

Responsibilities and Deliverables: As specified in the Simulator Requirements Derivation methodology, the Developer generates a series of requirements-related documents which are each subjected to the verification process with varying levels of intensity:

- Experiment Overview
- Functional Simulator Specification
- Simulator Approach
- Top-Level Requirements Specification
- Experiment Simulator Requirements Document (ESRD).

The emphasis of the verification process is on the ESRD, the end-product of the requirements definition phase, to ensure that the ESRD properly documents the requirements to achieve the objectives of the instructional Plan. The Verifier analyzes each of these documents and produces the Verification Analysis Report. The Verifier presents the results of his or her findings at the Simulator PRR and the PDR.

Methods: The focus of requirements verification is to analyze the ESRD for traceability, completeness, consistency, feasibility, and testability. In addition, the Verifier reviews all of the requirements-related documents for technical sufficiency and traceability. This analysis includes a reading of the document, the use of automated tools as appropriate, and trade studies. The methods employed differ for each objective as follows.

Traceability: The Verifier will examine each document to ensure the elements of each document are traceable from one document to the next, as highlighted in Figure 5-3. The developer of the documents provides the traceability information which is then reviewed by the Verifier. The Verifier examines the Experiment Overview to determine that each of its elements track back to data items in the experiment data base. The Functional Simulator specification is tied to the elements of the Lesson specification, Integrated Simulator Requirements, high-level Training Objectives, and PI Objectives. The Simulator Approach document is then examined to verify that it traces to the Functional Simulator and Experiment Overview. The Top-Level Requirements trace back to the Experiment Overview and Simulator Approach, and trace forward to the ESRD. The Verifier will examine the requirements to answer the following questions:

- a) Are the requirements sufficient to realize the original Instructional Plan objectives? A traceability matrix may be required to ensure sufficiency.
- b) Are all requirements traceable to the Instructional Plan? No extraneous requirements are allowed.

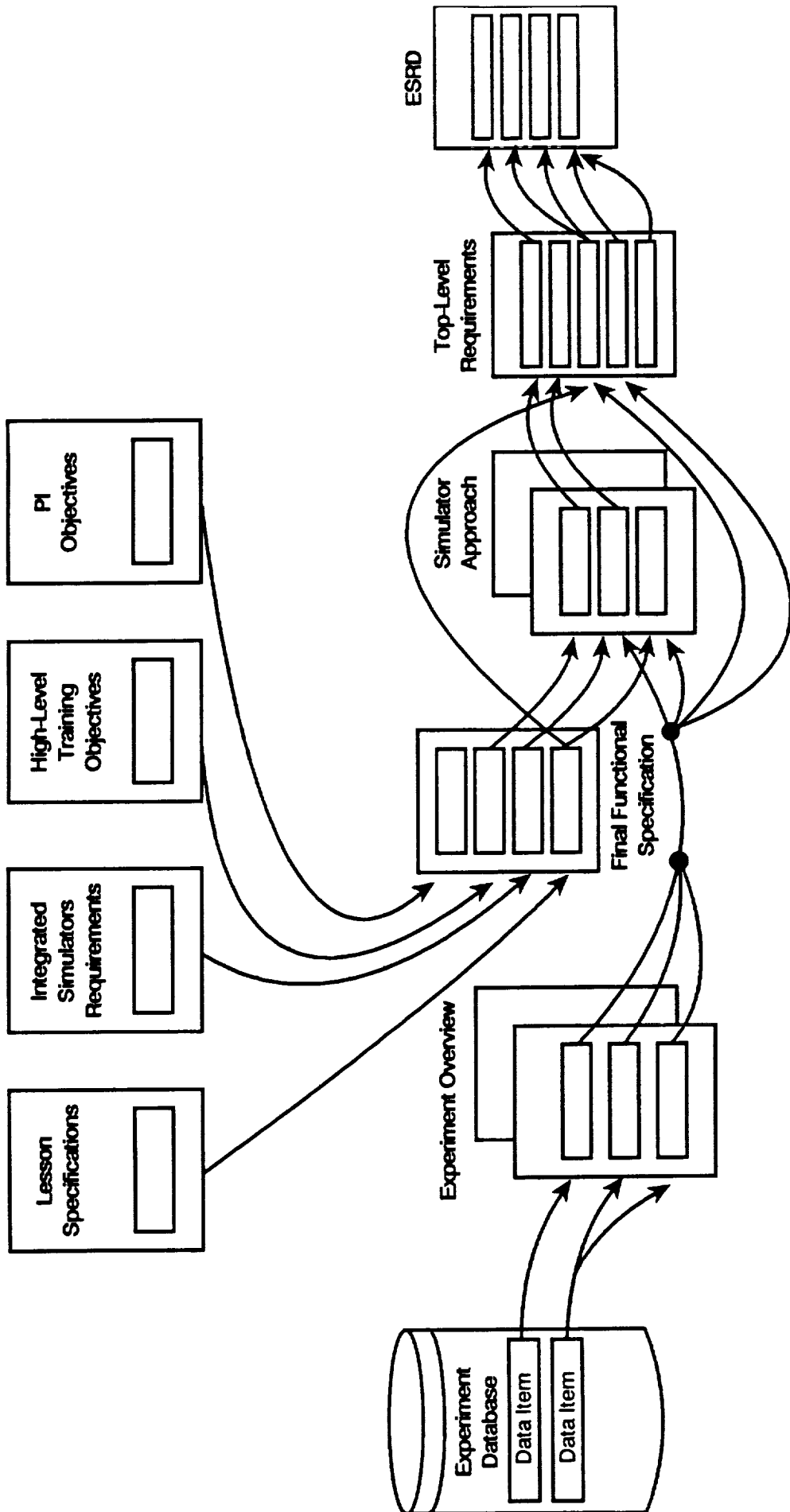


Figure 5-3. Simulator Requirements Verification Traceability

Completeness and Consistency Checks: The Verifier will examine each document to determine its overall technical adequacy and soundness. For the ESRD, the Verifier will use the checklists and manual cross-referencing to perform a more in-depth analysis for completeness and consistency. The Verifier will examine the requirements to answer the following questions:

- a) Are all functional requirements complete; i.e., no TBDs, no nonexistent references, no mission items?
- b) Are all performance requirements complete; for example, is a performance requirement stated wherever necessary?
- c) For each component, are the inputs, outputs, and processing requirements consistent and without ambiguity? All inputs have a source? All outputs have a sink?
- d) Are requirements stated in a logical, understandable, and traceable manner?
- e) Are all hardware interfaces identified?
- f) Are all software interfaces identified?
- g) Are all data base and data requirements clearly stated?
- h) Are all equations verified for correctness?
- i) Are user interface aspects adequately addressed?

Feasibility: The Verifier will identify any critical and/or high-risk elements of the simulator to be subjected to more in-depth feasibility studies. The Verifier will conduct a cost versus benefit analysis of the resources required to implement those requirements. Where appropriate, the Verifier will propose alternative methods of achieving the same training objectives with simpler technical solutions. The Verifier will also examine the requirements to ensure timing restraints and sizing resources can be met.

Testability: The Verifier will manually examine the ESRD for testability; that is, are all requirements specific, unambiguous, and quantitative wherever possible.

5.2.4 Design Verification

VERIFY THE SIMULATOR DESIGN DOCUMENTATION TO EVALUATE THAT THE DESIGNS ARE RESPONSIVE TO THE REQUIREMENTS AND DESCRIBE A TECHNICALLY ADEQUATE STRUCTURE FOR IMPLEMENTATION.

Purpose: The Verifier conducts the Simulator Design Verification activity to ensure that the specified designs:

- Represent a clear, consistent, and accurate translation of the requirements.
- Will serve as an appropriate baseline for coding.

The Verifier is to identify any inadequacies in the design. The Verifier does not have the job of attempting to redesign the product.

Responsibilities and Deliverables: The developers generate both a top-level preliminary design and a detailed design. The designated primary PTC simulator developer will provide an organized and integrated design structure to the Verifier, assuming various development organizations may be responsible for producing different portions of the training simulators. The Verifier analyzes each of the documents in turn and produces the Verification Analysis Report. The Verifier presents the results of his or her findings at the Simulator PDR and CDR.

Methods: The Verifier examines the design specification and uses manual analysis techniques, augmented with automated techniques where available, to answer the following questions:

- a) Does the design address all requirements as specified in the ESRD, including all updates to requirements? A traceability matrix may be required to ensure sufficiency.

- b) Is the design logical, understandable, and detailed enough to begin coding?
- c) Are all inputs and outputs correct?
- d) Are all algorithms correct?
- e) Is the data base architecture fully defined and logical?
- f) Are the internal and external interface designs sound?
- g) Are timing and sizing budgets established, and do they leave sufficient margin for growth?
- h) Have performance requirements been addressed properly in the design?

The Verifier will concentrate his or her energy on determining whether the entire simulation design structure will fit together into a cohesive training system. The Verifier will examine designs for simulators and flight equivalent equipment being supplied by a Principal Investigator to ensure interfaces and functionality are consistent with the overall design.

5.2.5 Implementation Verification

VERIFY THE IMPLEMENTED SIMULATOR COMPLIES WITH THE TECHNICAL DESIGN APPROACH.

Purpose: The purpose of Implementation Verification is to confirm that the as-built simulator complies with the technical design approach.

Responsibilities and Deliverables: The Simulator Developer produces unit-tested code and integrates the code into the trainer environment. Concurrent with the code production, the Verifier evaluates the code listings for errors, omissions, and violations of coding standards. The Verifier produces a Verification Analysis Report for in-progress input into the development activities.

Methods: The Verifier will use checklists, manual cross-referencing, and/or automated code analyzers to examine the code listings for technical correctness and adequacy. The Verifier will analyze both interim and final versions of the code as it is available. The Verifier will examine the code to answer the following questions.

- a) Each unit produces correct output for prescribed inputs?

- b) Arithmetic results are correct for nominal conditions?
- c) Minimum and maximum inputs are processed correctly?
- d) Scaling and data formatting is proper to realize correct precision and desired results?
- e) All error conditions are processed correctly?
- f) All branches are exercised?
- g) Timing restraints and resource allocations are mechanized properly?
- h) Any violations of programming standards?
- i) Any violations of prescribed code commenting standards?

5.3 Verification Testing

VERIFY THE IMPLEMENTED SIMULATOR FULFILLS THE SIMULATOR REQUIREMENTS BY PLANNING AND CONDUCTING INFORMAL "FREE-FORM" TESTING AND FORMAL ACCEPTANCE TEST PROCEDURES. CONDUCT THE SIMULATION ACCEPTANCE REVIEW.

5.3.1 Purpose

The purpose of Verification Testing is to plan and conduct acceptance tests to verify that the implemented software fulfills the simulator functional and performance requirements. The Verifier also performs informal "free-form" testing to verify the overall integrity of the system and to confirm that illegal activities and unusual combination of activities do not adversely affect the system.

5.3.2 Responsibilities and Deliverables

The Developer generates the software code according to the verified design baseline, develops unit-level test plans and procedures, conducts unit level tests, integrates the simulator system (including PI-developed simulators and flight equivalent equipment) into a coherent executable system. The Developer generates and conducts tests to demonstrate that each module of the implemented software fulfills the designs and the simulator requirements.

As shown in the TRDS program template, the Verifier develops an increment specific Acceptance Test Plan as an adjunct to the Master Generic Verification and Test Plan to describe any

increment specific testing needs, and presents the plan at the Simulation PRR. The Verifier develops Acceptance Test Procedures for the simulator system and summarizes that report at the CDR.

Upon delivery of the simulator system to the verification group, the Verifier executes the test procedures, generates Discrepancies Reports as appropriate, retests updated software, and generates a test summary. The Verifier generates and maintains Test Data Folders which contain detailed descriptions of test activities.

The Verifier then conducts the Simulation Acceptance Review (SAR). The Verifier presents the testing results at the SAR, and repeats a selected subset of the Acceptance Test for the reviewers. At the SAR, the PI is responsible to witness the demonstration of the tested simulators and comment on their accuracy and fidelity. The PI is responsible to witness the demonstration of the tested simulators and comment on their accuracy and fidelity. The PI is then encouraged to participate in a "free-form" hands-on test to perform any informal testing as desired. Proposed changes to the current baseline simulator requirements will be recorded, and only those changes considered mandatory for training will be given priority for implementation as directed by the project office.

5.3.3 Methods for Test Documentation Production

The Generic Master Verification and Test Plan, produced during the first Verification activity, is an expansion of this methodology and will define the top-level concepts and goals for each level of testing. Following the schedule defined in the TRDS program template, the Verifier produces the increment-specific Test Plan to describe additional testing concepts and goals as necessary to that increment. In the Acceptance Test Procedures documentation, the Verifier first defines and organizes the test cases to allow traceability from requirements to tests. The Verifier produces a traceability matrix to show that all functional and performance requirements in the ESRD are being verified by one or more test case. For each test case, the Verifier will document:

- The major capability under test
- The necessary test environment
- Required test inputs, including user actions and preset data values
- Method for observing test output (e.g., screen observation, data value extraction via test tool, etc.)
- Use of test tools to initialize and extract data values where appropriate
- Test acceptance criteria.

For each test case, the Verifier then defines test procedures at the level of keystroke entries, and describes specific inputs, actions, and outputs.

The Verifier establishes and maintains Test Data Folders which includes the test descriptions, traceability matrices of test cases to all requirements, the keystroke-level test procedures, a log of test results, a log of all incorporated software changes, retest, and results; and a log of open items.

5.3.4 Test Execution Activities

The actual execution of the verification testing is organized into three stages:

- (a) Increment-Independent Simulation Environment Testing
- (b) Informal "Free-Form" Testing
- (c) Simulator Acceptance Testing

(1) Increment-Independent Simulation Environment Testing: From time to time, the basic simulation environment provided by the PTC SCS will be modified and upgraded as authorized by approved Engineering Change Requests (ECRs). The Verifier will perform specific tests to verify those upgrades, and then perform regression tests as appropriate to assure those upgrades did not inject any undesirable side effects into the overall environment.

(2) Informal "Free-Form" Testing: Prior to the initiation of the formal acceptance testing, the Verifier tests the simulator in a "free-form" manner. The Verifier has the opportunity to informally checkout the overall soundness and integrity of the simulator system. This informal testing would include the entry of illegal commands and illegal combinations of legal commands to verify the overall adequacy of the simulators. The Verifier records any discovered anomalies on a Simulator Discrepancy Report form.

(3) Acceptance Testing: After the informal tests, the Verifier will execute the Acceptance Test Procedures in a controlled environment as defined in the Acceptance Test Plan. The Verifier will execute each test procedure and verify the actual output with the expected outputs as documented in the ATP. The test result of pass or fail is recorded in the Test Data folder. Any discrepancies are recorded on the Simulator Discrepancy Report form, and forwarded to the project office for resolution. As appropriate, the Verifier will use automated test tools to create the

test environment, execute the test procedures, obtain the required output data, compare the actual outputs with expected outputs, and record the results (pass or fail) of the test execution. During the definition and execution of the test procedures, the Verifier will consider the following checklist for examining the outputs:

- All inputs are accepted and produce correct outputs?
- All limits of legal input data are handled properly?
- All screen displays are formatted correctly?
- All data files are updated correctly?
- All error conditions are tested? All error handling is performed properly?
- Algorithms and models produce the correct results?
- Initialization activities are properly implemented?

5.4 Summary

In summary, the Verification process consists of the following activities:

(a) Produce the Generic Master Verification and Test Plan as an increment-independent verification guide for the development of all training systems.

(b) Perform Specification Verification at each stage of the development process to ensure the output of each stage is verified and baselined prior to proceeding with the next stage. The Verifier presents the results at each of the major program reviews. The five stages of Specification Verification are:

- (1) Training Objectives Verification
- (2) Instructional Plan Verification
- (3) Simulator Requirements
- (4) Design Verification
- (5) Implementation Verification.

(c) Plan and conduct Verification Testing to demonstrate that the implemented simulators fulfill the simulator requirements. The Verifier concludes the Verification and Test process with the Simulation Acceptance Review.

6.0 TRAINING SYSTEM VALIDATION METHODOLOGY

The PTC Training Development Validation Methodology defines a process to ensure that the total training system developed for each Space Station experiment fulfills its overall training objectives. Unlike Verification, which is concerned with a simulator's individual capabilities, Validation is a process of evaluating a simulator's integrated ability to fulfill its purpose, that is to provide training. In addition to simulator or hands-on media training, the Validation process involves evaluation of the academic training which will be provided as part of the total training offered for each experiment. Verification and Validation have been described elsewhere as intertwined activities throughout the development process. They both use the same tools and analyze the same data items. For our purposes, however, Validation will be a separate activity starting later in the development process when the piece parts have been integrated and the final product is to be evaluated.

Validation is conducted in a more realistic environment (such as closer to the actual conditions of use) than Verification is conducted. Also, Validation involves the integrated use of ideally, all supporting materials and all personnel positions required for normal training, to validate that the training system configuration will actually work as planned. The term "training system" (for this discussion), refers to the entire collection of instructional materials, simulators, scripts, training personnel and lessons, both academic and "hands-on," used to implement all stages of training for a particular experiment.

Validation will be performed by either the same people who are performing Verification, or at least by a group detached from the development crew. This Validation group, herein known as the Validator, provides NASA with an objective and independent perspective to assess the training system capability to meet its objectives.

Training systems should be validated by comparing them with the training objectives and functional requirements from which they were designed. These criteria are one step removed from the specific implementation details which were the focus of Verification and relate directly to the various training functions of the system.

The Validation procedure therefore, will consider all stages of training from familiarization to integrated mission simulations. For example, the academic training objectives will be used to validate CBT courseware and classroom lessons, while hands-on media Functional Specifications will be applied to simulator training validation. The Validation process will consider a wide

variety of inputs, such as JSC concerns, PI-provided training objectives, and integrated training functions which were factored into the Functional Specification before it was finalized. The Validation process begins with the production of Test Plans which will be performed to validate all training development end-products. A Test Plan is defined as a set of directions for conducting a test which state conditions, methods, and procedures to be used. As shown in the TRDS Template given in the Program Concept, Test Plan development for academic instruction begins about midway through the detailed design phase, though it could actually start as soon as the appropriate academic Lesson Specifications have been verified. The Lesson Specifications define the lessons to be produced, and so are necessary as guides for Test Plan formulation in lieu of the actual lessons though they are not directly used as Validation criteria.

Test Plan development for hands-on or simulator instruction begins after simulator CDR, when instructional materials supportive of simulator training become available. Like academic Test Plan development, this effort could start sooner, in this case, as soon as finalized hands-on media Lesson Specifications have been approved. The Functional Specifications define the simulator functionality necessary to meet allocated training objectives. The hands-on Lesson Specifications define the supporting lessons and instructional materials which will be used in conjunction with the simulator to provide hands-on training.

Test Plans will be used to validate each simulator, each lesson, and to evaluate the overall integrity of the provided training system. Validation of Academic Instruction will commence as soon as the academic lessons, courseware, and supportive materials are complete, but before classroom or CBT training is scheduled to start. Validation procedures for hands-on training will be conducted for each simulator at its Simulator Training Acceptance Review (STAR). See Figure 6-1 for a graphical representation of this scheduling.

Once a training system has been validated, and pronounced Ready For Training, further validation activities will continue throughout the training cycle. Ongoing Validation will evaluate student performance in various ways to ensure that effective training occurs, and to detect and diagnose problems with the hardware or with the training regimen. Corrective changes will be recommended both for current training, and for the training development methodology.

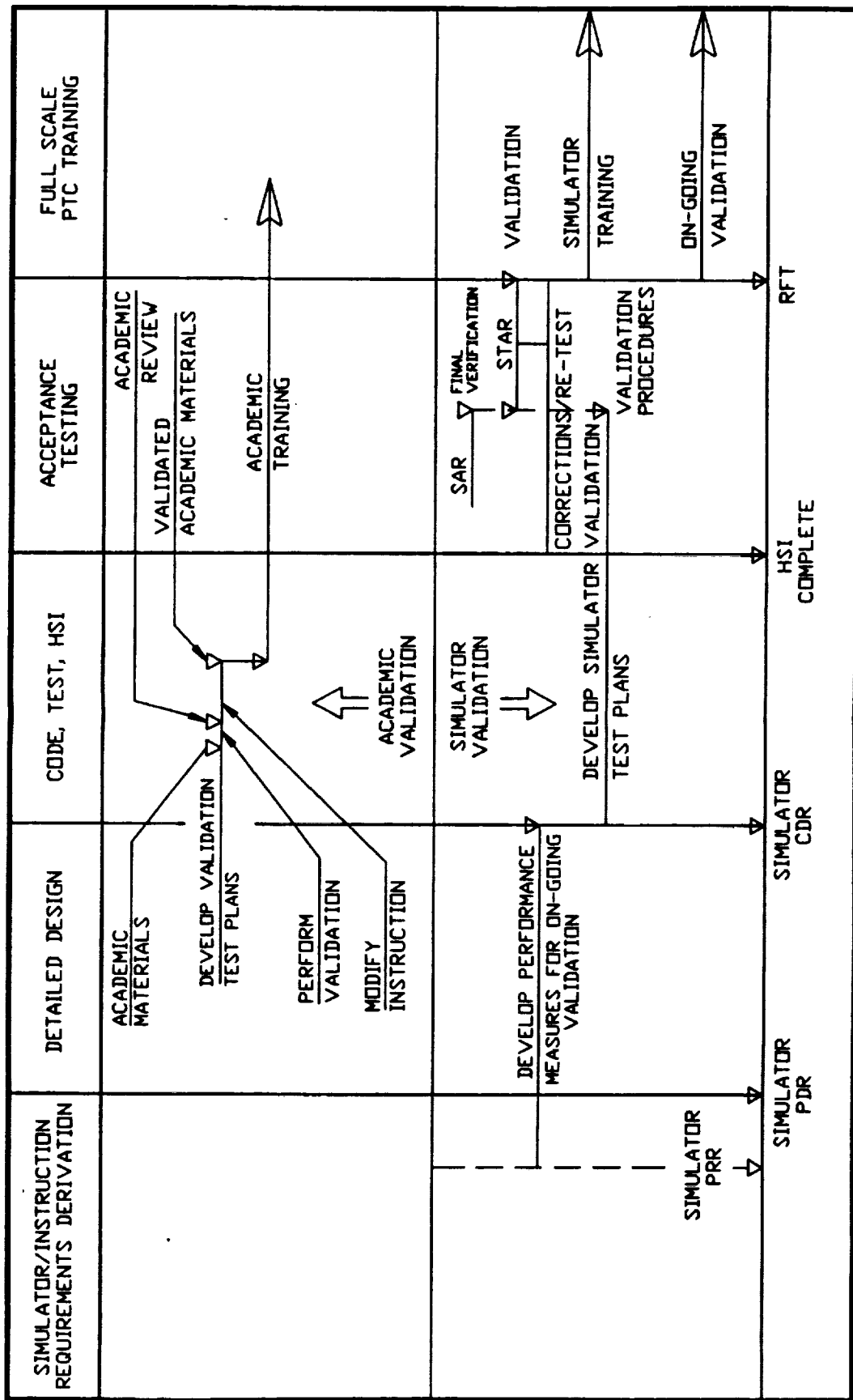


FIGURE 6-1. SIMPLIFIED TRAINING PROGRAM SCHEDULE FOR VALIDATION

6.1 Academic Instructional Validation

This is where the lessons and instructional materials designed to fulfill academic training objectives are validated in actual use with academic media such as classrooms or CBT terminals. Whereas Verification will have been performed on the Lesson Specifications from which these academic end-products were designed, Validation testing will ensure that the various instructional elements in combination will meet their parent training objectives. Since the training objectives were derived from the tasks to be performed by different crew members, their use as validation criteria will ensure that the different training needs of the various flight and ground crews will be met by the proposed curriculum.

6.1.1 Academic Instruction Test Plans

FORMULATE TEST PLANS FOR VALIDATION OF ACADEMIC INSTRUCTION

Test Plans for the conduct of academic instruction validation will be assembled at some time following the availability of Lesson Specifications for the instruction to be validated. Lesson Specifications contain the parent training objectives to be fulfilled by the lesson. They also contain overall instructional strategies and a description of the instructional materials required to conduct the lesson. Therefore, since the academic training objectives comprise the Validation criteria, the Lesson Specifications will serve as an excellent guide for Test Plan production.

In constructing a Test Plan for a specific lesson, the academic Lesson Specification will provide a list of all materials to be evaluated. This would include workbooks, slides, courseware, exhibits, and any other material used in the instruction. The Test Plan will be organized in terms of how to evaluate each instructional item or combination of items. For example, the Test Plan might include a section for evaluation of a workbook. This section would:

- (a) Specify the method of examination. (In this case, reading or reviewing would be appropriate.)
- (b) List all points or topics to be covered by the workbook. (These points would be derived directly from the training objectives.)
- (c) Discuss the tasks to be performed by the Validator while reviewing the workbook. (These would be procedures

such as to check for conflicts and completeness in the presentation material with respect to the list of topics in "b" above.)

(d) Pose questions to the Validator relating to the sufficiency of the material for preparing a student to satisfy the performance measurement criteria listed with each objective. An example question might be: "Would completion of this chapter enable the student to explain all the functions of a viriometer, without assistance?" These questions would be derived directly from the performance-driven Criterion Objective and Diagnostic Tests developed for each objective.

The end of this section would include a Validation Test Matrix to identify training objectives or requirements and their corresponding validation tests. The Validator will fill out the matrix while performing validation on the workbook.

A Test Plan section for CBT courseware would be constructed in a similar fashion. The method of examination would be to key through the material using the CBT terminal. There would be a list of points derived from training objectives which would be validated in the courseware by performing the procedural check recommended. A set of questions relating to performance measurement criteria would be asked, and a Validation Matrix would be provided.

Validating lessons or parts of lessons which use methods of instruction involving other people, such as a lecture or classroom situation, requires a slightly different approach. The Test Plan section might include:

a) Instructions for the setup of a live enactment of the instructional situation. A surrogate or actual instructor would be used for the teacher while the Validator would assume the role of the student.

b) A list of points or topics to be covered, derived from training objectives.

c) Instructions on the procedures for evaluating the lesson, such as to check for conflicts in the material's presentation, completeness with respect to the points to be covered, and a subjective evaluation of the effectiveness of the instructional strategies employed in the presentation. This last would include impressions on the effectiveness of slides, exhibits, flip-charts or any other instructional aid employed.

- d) Questions relating to performance measurement criteria, as discussed above in the workbook and courseware examples.
- e) Directions for how the enactment may be altered or abbreviated for Validation testing in ways which will not prove detrimental to Validation purposes.

Alternatively, if time or resources prohibit a live enactment, then the lesson could be validated simply by inspection of the class materials in the same manner in which the workbook mentioned above was validated. In either case, a Validation Matrix should be provided to ensure that the Validator systematically tests for adherence to training requirements.

For all Test Plans, no matter what the format, the important considerations are that they define specific criteria for identifying and correcting deficiencies, certify that the training system will satisfy training objectives, and relate back to performance measurement criteria previously defined -- which means that the Test Plan must be able to provide some assurance that a person taking the course will be able to meet the criterion objective and diagnostic tests.

6.1.2 Conduct of Academic Validation

PERFORM ACADEMIC VALIDATION TESTING

Academic instruction was verified at the Lesson Specification level to ensure that the lessons would contain the information necessary to accomplish the training objectives. The resultant lessons, academic media, and instructional materials will be validated in use to ensure the same thing. This validation will be accomplished by noting conflicts within developed lessons during presentation which were not apparent in the lesson planning stage. Workbooks and other materials will be checked for completeness and lack of conflicts, no missing references, or parts. Lesson presentations will be compared with the parent training objectives to validate that they have been satisfied in a clear and orderly manner. Using a Validation Matrix, this comparison will ensure that an accounting is made of the lesson for satisfaction of all objectives, no missing points or explanations.

General Validation methods will range from reviewing instructional materials to actually running through a class presentation with the instructional materials, exhibits, slides, etc. These simulated training scenarios may be condensed or abbreviated, but they should provide ample opportunity to evaluate the training development products in use, against the

appropriate training objectives. For Validation, it will not be necessary to actually teach students the material (though you certainly could), but the testing must enable the Validator to make subjective decisions about its effectiveness.

REPORT VALIDATION RESULTS

After the Validator has examined all academic training materials and conducted live validation tests, he or she will prepare an Academic Instruction Report. This report will highlight issues, problems, and potential solutions, and will be presented at an Academic Instruction Review. After the Review, the Validator will document the approved changes in formal Engineering Change Requests (ECRs) which will be submitted to the proper organization to implement the change. The schedule and milestones for this validation process are documented in the Program Concept TRDS template.

6.2 Hands-On Media Validation (Including Simulators)

Hands-On Media Validation is the process of ensuring that the various elements which have been developed for hands-on training provide the proper functionality to support all training objectives and planned use. These elements are comprised of simulator hardware and software, support equipment, training scripts, lesson plans, and any other aids required to facilitate hands-on training. In contrast to Verification, which tests instructional materials and simulator hardware and software for their individual characteristics, Validation will ensure that all of the elements work in combination to provide the required training.

The hands-on media Functional Specification for the training simulator and higher level hands-on training objectives will be the primary criteria for hands-on training objectives, which in turn were derived from the tasks performed by different flight or ground crew members. Therefore, like the academic training objectives used for validations of academic instruction, the use of the Functional Specification and hands-on training objectives as validation criteria for hands-on instruction will ensure that the different training needs of the various flight and ground crew will be met by the simulator functionality.

6.2.1 Hands-On Media Test Plans

**FORMULATE TEST PLANS FOR VALIDATION OF
HANDS-ON INSTRUCTION**

Test Plans for the conduct of simulator instruction validation will be assembled at some time following the availability of either simulator Functional Specifications or the Lesson Specifications for that simulator. Ideally, both would be available when Test Plans are formulated since the tests will be based on an integrated scenario involving personnel, scripts, and the simulator.

There will be a group of Test Plans written for each simulator. Each Test Plan will be built around a specific lesson as described by its Lesson Specification. Some lessons will involve one flight crew member interacting with a single experiment. Others will involve a ground controller and/or additional flight crew members. Still others will involve multiple experiments interacting, or simultaneously operating in a timeline environment. For each case, the Test Plan must specify the required personnel and simulator configuration. The lesson scenario described by the Plan may be simplified for Validation if possible, but it must demonstrate the simulator configuration's capability to satisfy all criteria. The criteria to validate this capability will be the simulator functional requirements and the training objectives taken from the simulator Functional Specification and the higher level training objectives respectively.

In constructing a Test Plan for a specific lesson, the Lesson Specification will provide an outline of tasks to be performed, a description of the instructor interaction to be provided, and a list of the parent training objectives to be fulfilled by that particular lesson. The Functional Specification will provide descriptions of the requisite trainer functionality, traceable to hands-on training objectives. Using these inputs, the Test Plan will be organized to exercise the simulator configuration in such a way as to systematically demonstrate the simulator's capability to satisfy the higher level training objectives targeted by the Lesson Specification, and the simulator's capability to satisfy as many of the training objectives as possible for which the simulator was designed, within the constraints of the lesson.

Typically the most basic scenario (stand-alone experiment operation) would be tested first. In this case, the overall Lesson objective might be (roughly) to "perform the experiment."

The Test Plan would then be developed to exercise the simulator's capabilities as much as possible within the scope of the lesson. In most instances, this basic scenario would provide opportunity to validate most capabilities of a particular simulator configuration. However, objectives such as to "conduct interactive operations with experiment XYZ" could not be validated under this scenario (and this simulator configuration) but would be covered under the Test Plan for the lesson concerned with those interactive activities. For that test, it would not be necessary to re-validate any capabilities, but simply those which had not yet been tested within the constraints of the lesson and the simulator configuration.

Each Test Plan then, will include:

- (a) A description of the test scenario in terms of the personnel required and the simulator configuration (stand-alone, integrated, etc.)
- (b) A listing of all tasks (or script) to be performed by the flight and ground crew members, and by the instructor during the test. This will be based on actual Crew Procedures and training scripts (if available), but will be adapted to systematically demonstrate the simulator configuration's capability to meet specific training objectives and functionality requirements.
- (c) A list of test criteria both general and specific adapted from the Criterion Objective and Diagnostic Tests for the training objectives being addressed. The Validator will use this criteria in deciding if the training objectives can be met by the simulator configuration.
- (d) Directions for how the test scenario may be altered or abbreviated in ways which would not harm Validation.
- (e) A Validation Test Matrix to identify training objectives or requirements and their corresponding Validation tests. The Validator will fill out the matrix while observing the Test Plan scenario.

Since many of the simulator configurations to be tested will require other experiment simulators, there must be constructed a Master Validation Test Plan to coordinate the sequencing of all Validation tests. This Plan must consider the development schedules of the various experiment simulators so that when integrated testing is scheduled, all required resources such as other simulators) will be available. Additionally, the Plan must be coordinated with the PTC Increment Training Schedule so that simulators will not be used for training purposes for which they have not yet been validated.

6.2.2 Conduct of Hands-On Media Validation

PERFORM HANDS-ON VALIDATION TESTING

Experiment simulators will be validated by running a training scenario or Test Plan adapted from actual flight or training materials (if available). Testing should be conducted in a realistic operating environment where hardware, environmental, and personnel effects are in the loop. The lesson plans, Crew Procedures, or training scripts around which each Test Plan is built may be modified and abbreviated for expediency, but the resultant Validation Procedure must still exercise the entire training environment in a way calculated to demonstrate that all training objectives can be met by the simulator configuration. The procedure must also demonstrate that the system is feasible from the operational standpoint of the students and instructors. The Validation scenario should be monitored for problems in execution, such as combinations of cues which perform adequately on an individual basis, but do not interact correctly. Instructor functions should be scrutinized to discover those which do not work well during an actual simulation. Obviously, feedback from the scenario participants will be a primary, though not exclusive, input to this type of evaluation. A primary purpose of simulator Validation is to demonstrate the proper compatibility between the hardware, software, and simulator instructional materials used for training.

After individual simulators have been validated, they must then be operated simultaneously and/or interactively with each other just as they will be during actual increment operations. These later tests will be guided by the Master Verification Test Plan which coordinates the testing of the higher level training objectives. During these test scenarios, Validation scripts will be followed. These are calculated to demonstrate the capability of integrated simulator groups to train higher level objectives such as team coordination, timeline procedures validation, and communications protocol. Often it will be possible to combine the Test Plans for two or more simulators when operated together in the same test in order to validate all of the simulators at once.

Validation testing for single and integrated simulator operations will be performed as part of the Simulator Training Acceptance Review (STAR). The purpose of this Review will be to demonstrate the capability of the simulators as training tools. Following the STAR, the simulators are considered operational and ready for use by training personnel. In practice, a simulator will be

usable to the extent to which it has been tested. For example, if stand-alone testing on a simulator was performed successfully, but higher level testing was postponed for scheduling reasons, the simulator could be used in stand-alone training, even if integrated Validation was not yet complete.

REPORT VALIDATION RESULTS

During the STAR, problems or perceived needs for new requirements are noted and discussed with the STAR team. Proposed changes to the simulator baseline will be discussed, and only those changes deemed mandatory for training will be documented by the Validator as ECRs. After the Review, the Validator will submit them to the proper organization to implement the change. Depending on the nature of the changes and the program schedule, the Validator may provisionally approve the simulator for training while the ECRs are being cleared, or he or she might withhold approval until necessary changes can be implemented. The schedule and milestones for this validation process are documented in the Program Concept TRDS Template.

6.3 Ongoing Validation

After determining (through Validation) that the correct training systems have been designed and built, it is desirable to validate on a continual basis that the training systems are providing correct training. This will afford a degree of quality control for the immediate training process as well as to generate recommendations for improvement of the training development system for future training. Rather than focusing on training design criteria, as does the initial validations, Ongoing Validation will detect problems by evaluating student performance.

6.3.1 Performance Measures for Ongoing Validation

DERIVE PERFORMANCE MEASURES WHICH CAN BE USED TO EVALUATE TRAINING EFFECTIVENESS AND DIAGNOSE TRAINING PROBLEMS

Efforts for Ongoing Validation will begin around the time that Lesson Specifications are being assembled. The Lesson Specifications will include a Performance Evaluation Plan which contains tests for each training objective and Performance

Measures for overall training effectiveness. Part of the Ongoing Validation responsibility will be to help derive some of these measures (see Section 3.2.3, Lesson Specifications). Appendix B, Metrics, discusses various types of training measures, their purposes, and issues surrounding their selection and use.

In general, the training development effort will concentrate on deriving measures which will be used to evaluate individual student's progress and predict future performance. The Validation team on the other hand, will concentrate on measures which will help diagnose training problems and determine training effectiveness. Obviously, there will be considerable overlap because single measure often can serve multiple purposes. Ongoing Validation will also be concerned with academic as well as with hands-on media training. Problems with presentation and delivery of instructional material in a classroom or at a CBT terminal can occur as readily as with an experiment simulator. However, since the academic setting lacks the complex man-machine interaction of simulator training, it is expected that less attention will be focused there.

6.3.2 Conduct of Ongoing Validation

EVALUATE STUDENT PERFORMANCE ACCORDING TO DERIVED
PERFORMANCE MEASURES

MODIFY TRAINING OR THE TRAINING METHODOLOGY BASED ON
PERFORMANCE EVALUATION

Over a 30-year lifetime, the training development system will undergo changes and hopefully, improvements. Some of these changes will be occasioned by new technology, and will affect the development system hardware and software. Other changes will be recommended by Ongoing Validation activities and will affect the development system methodology.

Ongoing Validation will be conducted by analyzing the performance of the PTC training systems through use of the derived performance measures. These measures, while directly measuring student performance in various ways, will reflect on the systems which provide the training as well. Other inputs to the training system analysis process will include feedback from instructors and students on training problems.

Metrics developed for Ongoing Validation will measure both individual and group task proficiencies. Since development of task proficiency can be regarded as a primary purpose of the

training system, measurements of task proficiency may be used to determine training system effectiveness, and by further extension, the effectiveness of the training development methodology. Concepts such as training effectiveness and transfer of training represent the indirect effects, rather than the products of training; therefore, there are no direct measures for training parameters which are measurable such as task proficiency. In order to calculate values for effectiveness (either of the training system or the development system) proficiency measures must be combined with other variables, such as the time required for skill acquisition, training development time, training development cost, cost to conduct training, etc.

Once overall effectiveness values for delivered training are determined, they may be used to optimize specific facets of the development process. This will be more difficult than optimizing a specific training system since the development system is one level of abstraction removed from the training system. What may be necessary is a direct comparison of training outcomes using two alternative development systems. For example, two methods for determining minimum simulator fidelity levels could be contrasted by comparing training effectiveness values derived from two similar trainers. Each trainer would have to be developed under a methodology differing only in the factor under study. In this way, judgments may be reached concerning alternative training and training development methods.

The proceeding discussion implies that, to improve the system, deliberate efforts must be made to collect empirical results and interpret them in accordance with programmatic imperatives (resource utilization). These results are then traced back to their specific causative factors by means of an express testing regimen. If a more direct feedback of corrective inputs is desired, then less rigorous methods may be used with a concomitant loss of certainty and specificity of conclusions. For example, user comments, as previously mentioned, could be collected and intuitively linked with specific development processes which would then be modified accordingly.

Problems in training discovered through Ongoing Validation will be documented in an ECR along with the change(s) recommended for its solution, and submitted to program management. Program Management will evaluate the ECRs and if approved, will route them to the proper organization(s) for action. These changes could involve devices and materials currently used for training, and/or could affect the training development methodology. All completed changes will be reviewed by Ongoing Validation.

6.4 Engineering Change Requests

Engineering Change Requests (ECRs) will be used to document suggested changes to simulators, academic media, or instructional materials. ECRs will be generated during the Validation period in response to problems found during Academic, Simulator, or Ongoing Validation. They will be submitted to project management who will forward them to the responsible organization for action. A log of ECRs will be maintained and after the change is made, the Validator will be responsible for verifying the modification as well as the requisite changes to training development documentation.

The ECR form must contain the following information:

- (a) Name of Originator (Phone, Organization, etc.)
- (b) Identification of simulator, lesson, software module, etc., where change must be made
- (c) Description of the change
- (d) Rationale for the change
- (e) Development documents and records upstream of the change which must be modified
- (f) Approval Block
- (g) Completion Block (affirms change was made, notes).

The organization responsible for implementing the change will initiate a two-pronged action. First, the change will be made upon approval, without delay. Second, all documentation upstream of the change will be updated as necessary to preserve a logical development flow. The ECR will remain an open change item until ALL documentation has been revised. This approach will ensure that changes are implemented as swiftly as possible, while preserving the integrity of training documentation.

7.0 SIMULATOR FIDELITY DEFINITIONS

On November 15, 1988, NASA Space Station Freedom training planning groups at JSC and MSFC agreed on a classification system for training simulators. The purpose of this system is to establish a common nomenclature between Space Station training groups to describe training devices in terms of their fidelity and functionality. The term "training device" in this regard refers interchangeably to trainers, simulators, or mockups. Thus, the spectrum of devices considered range from primitive representations of physical devices up to the actual flight or ground equipment whose use is to be trained.

The purpose of establishing a common nomenclature is to eliminate confusion over terminology when discussing the training devices to be developed and utilized for Space Station Freedom training. The intent is not to provide multi-variate device descriptions for specifications, but rather generalized "ballpark" designators suitable for use in common communications, and sufficient for top level resource planning purposes.

7.1 Terminology

Simulator/trainer/mockup: An assembly of hardware alone, or hardware and software in combination, configured to resemble some aspect of a flight element or piece of ground equipment.

Functionality: The degree of exactness of replication of the stimuli and the responses to those stimuli by the simulator/trainer/mockup relative to the original article.

Class: Appearance, tolerance, and composition of a simulator/trainer/mockup as it relates to the original article.

The classification description for a training device is two-dimensional, consisting of "Class" and "Functionality" as described above. Most of the training devices considered for Space Station Freedom training can be represented by pairs of variables corresponding to various values or degrees of the two qualities. Note that the two qualities each represent aspects of fidelity and functionality. "Functionality" as defined above, also incorporates the fidelity of a simulator's functional aspects, while "Class" basically represents a simulator's physical fidelity with respect to the original article. The classification system can be summarized in the form of the matrix shown on the following page.

Functionality	F	A	B	C
Class	Flight Type	Functionally Active	Operable	Static
Flight-Type	F-Flight Equipment Downgraded for Training	N/A	N/A	N/A
I. Flight assy. toler. similar material exact configuration	N/A	I.A	I.B	I.C
II. Relaxed assy. toler. mixed material approximate config.	N/A	II.A	II.B	II.C
III. Approx. dimensions optional material approximate config.	N/A	III. A	III.B	III.C

Table 7-1. Simulator/Trainer/Mockup Classification Matrix

As can be seen in the above matrix, there are four levels of functionality, and four classes of hardware which can be separately designated by the classification system.

7.2 Functionality

Four levels of functionality are required to fully address simulators/trainers/mockups:

Flight-type: The capability of utilizing de-rated actual flight or ground hardware to replicate the stimuli, processes, and responses of the original article. Flight-type capabilities provide simulated flight data and communications with appropriate transmission protocols.

Functionally active: The capability of functionally replicating the stimuli, processes, and responses of the original article. Functionally active capabilities provide simulated data and communications but need not use the same transmission protocols.

Operable: The capability of functionally replicating the stimuli and responses of the original article with limited process modeling. Data and communications are provided only to student and instructor.

Static: No active stimuli, processes, or responses.

7.3 Hardware Classes

In addition to flight-type hardware, three Classes of hardware are required to address the appearance, tolerances, and composition of simulators/trainers/mockups:

Class I

Flight Assembly Tolerance: Conforms to flight (or ground) article dimensions, but is not flight qualified.

Similar Materials: Materials are of same family and characteristics, but are not necessarily the same grade.

Exact Configuration: Appearance is like flight article in all aspects.

Class I hardware is typically used for crew (or ground) training or for engineering verification exercises.

Class II

Relaxed Assembly Tolerance: Not held to flight specifications; margins to be specified by requirements documents.

Mixed Materials: Materials meet general characteristics of flight article and optimally support the intended function, but need not be of the same family, grade, or specification.

Approximate Configuration: Appearance is similar to flight article (size, shape, color, orientation, location, etc.)

Class II hardware is typically used for crew (or ground) training or for design development purposes.

Class III

Approximate Dimensions: Anticipated volumetric approximation.

Optional Materials: Materials support facility objective.

Configuration: Appearance to depict design or anticipated concept.

Class III hardware is typically used for concept formulation or for preliminary layout. It is also used for portions of a training facility that do not require active student operations and would otherwise remain void. Example: a module window that crew training need not address.

7.4 Planned Experiment Simulator Usage at the PTC

The following are brief descriptions of the broad categories of payload training to be performed at the PTC, together with the simulator types most applicable to those training categories.

Experiment familiarization/Science background acquisition: This training will be performed as needed for the particular flight and ground personnel who are training to operate a particular experiment. The purposes of this training are to

- 1) provide a general background on the scientific basis for particular experiments
- 2) provide a top level treatment of the specific nature of an experiment and a basic understanding of its operation

Simulators: This training would be typically conducted in a classroom situation or with CBT courseware, neither of which falls within the scope of the simulator fidelity matrix. However, classroom training may utilize experiment exhibits or mockups as teaching aids. These aids would typically fall under simulator fidelity classifications III.C, or II.C.

Individual experiment operations: This training involves basic operations associated with the experiment as a stand-alone activity, whether accomplished through hands-on or academic media. The training may be oriented towards flight and/or ground controller activities. The purpose of this training is to teach basic procedural skills and knowledges necessary for primary

operations of the individual experiment either from the ground, or at the Station.

Simulators: The training objectives associated with this type of training may be partially satisfied by CBT courseware which is outside the scope of the simulator classification matrix. They may also be partially or wholly satisfied by static mockups, or stand-alone experiment simulators of limited operability. Therefore, the range of simulator types which could provide this training (depending on the specific training requirements) would be types II.C, or higher.

Integrated experiment operations within a module: This is training for activities concerned with the simultaneous and possibly interactive operation of multiple experiments within one module such as the US Lab. Rather than concentrating solely on the procedural aspects of individual experiment operations, this training will provide coverage of the mainstream skills, interpretive knowledges, and decision-making necessary for each experiment's operation. It will include the organizational problems of managing simultaneous experiments, timeline validation, and communication skills and protocol.

This type of training will require the greatest degree of functionality and cue fidelity with respect to the experiment processes. At the same time, since the interactive aspects of simultaneously operating experiments will also be trained, the simulators must provide data and command/response communications as well. Therefore, applicable simulator types would include type I.A, or F - Flight-type.

Consolidated experiment operations: This type of training is designed to teach the cooperative and communication skills necessary to coordinate payload operations in a specific increment for experiments located in several modules. The training exercise would involve most or all payloads for a given increment. This level of training will focus on teamwork skills, and the validation of operational procedures rather than on the accomplishment of individual experiment objectives. Similarly, whole station training will also involve station-wide coordination and communication skills, and use most or all of an increment's payloads, but will focus on core Space Station Freedom systems, rather than payloads. Resource allocations and basic station-keeping will be emphasized.

Since both types of training will be focused on higher-level organizational objectives, the simulators will not need the highest levels of physical fidelity. On the other hand, since training operations will involve interactions with other simulators and other training facilities, data and communications

to/from each simulator will be needed. Therefore, applicable simulator types would include type II.A, or higher.

8.0 REFERENCES

The following documents and reports were used for the PTMS:

Airborne Systems Software Acquisition Engineering Guidebook for Verification, Validation, and Certification. TRW document produced for Air Force System Command under Contract F33657-76-C-0677. September 1978.

Course Developers Course. Document No. CDC-107, Air Force. June 6, 1983.

Experiment Simulator Requirements and Simulator Design Development Handbook. Essex Corporation, JA-711. 1987.

Fleishman, Edwin A. and Quaintance, Marilyn K. Taxonomies of Human Performance. Academic Press, Inc. 1984.

Instructional System Development. Air Force Manual AFM 50-2. July 15, 1986.

Martin, James and Oxman, Steven. Building Expert Systems. Prentice Hall. 1988.

Military Specification Requirements for Training System Development. MIL-T-29053B. June 15, 1981.

Payload Crew Training Complex (PCTC) Experiment Simulator Requirements and Simulator Design Development Handbook. Teledyne Brown Engineering and Essex Corporation document produced for NASA/MSFC under Contract NAS8-32712. July 15, 1987.

Position Paper on Case Tools For PCTC. Dave Scott/NTI. October 13, 1988.

Spacelab Flight Software Test Plan. McDonnell Douglas Document MDC W5460 produced under subcontract by TRW for NASA/MSFC. July 1989.

Space Station Crew Selection Criteria, Revision A. Dave Walker/NASA. October 25, 1988.

"Verifying and Validating Software Requirements and Design Specifications." Software Magazine, IEEE Computer Society. January 1984.

APPENDIX A

Impact of Trainees' Group Characteristics on Payload Training

APPENDIX A

PTMS ISSUE OR DESIGN GOAL

Issue Title: Impact of Trainees' Group Characteristics on Payload Training Instructional Strategies

Issue No: I-14

Revision: 0

Date: 11/22/89

Assumptions

The primary responsibility of the TRDS is to develop training and trainers for the instruction of payload operations. Trainees are composed of the flight crew and the POIC cadre.

Screening of applicants for flight and ground payload crew positions is not a designated function of the TRDS.

Discussion and Rationale

When developing an instructional program, it is necessary to consider the existing skills, knowledge, and for some requirements, the psychological characteristics of the incoming trainees. In some cases, a preferred trainee profile and an applicant screening process must be developed which is based on the demands of the job for which training is to be conducted. Alternative means of obtaining qualified applicants should be considered. It may even be possible to obtain applicants who have already acquired some or all of the requisite qualifications. In all cases, the Instructional Plan must account for the incoming trainees' proficiencies and deficiencies when determining what to teach, and how to teach it. Course content as well must be sensitive to the initial knowledge and skill levels of the trainees.

For Space Station Payload Training, it is assumed that incoming trainees will be pre-selected for their tasks and that no profile development or applicant screening activity will be required by the training development function. In deriving Instructional Plans, however, it is important to be familiar with the anticipated learner characteristics, because they will be a factor in deciding:

What To Teach: If for example, most trainees for a ground controller position will already be familiarized with generic POIC console operations, then that course of study may be excluded.

Extent of Instruction: Higher aptitude, or more educated trainees may be expected to absorb a curriculum faster, and with

less repetition than others. Trainees who are highly motivated will require less reinforcement during training.

How to Teach: Motivation and ability to accept responsibility will affect the degree of independence given to students within the curriculum. Students with learning anxieties benefit from more structured presentations where there is less of a burden to provide their own learning structure.

This study will explore what is known about the preferred profile of flight and ground crews for payload operations. This profile will be used to make preliminary recommendations for instructional planning.

Inputs

Issues in Training Device Design and Prediction; Seven, Babbit, Muckler, 1988, Essex Corporation

Draft Space Station Crew Selection Criteria, Rev A.; Dave Walker, 1988

Space Flight Crew Selection Criteria; Dave Walker, 1988

Interviews with Andy McClendon/TBE, and Lynn Baker/TBE, Huntsville

Conclusions/Solutions

Space Station policies for flight and ground crew qualifications are still evolving. When leavened with common sense, however, enough information is available to allow some preliminary judgements to be made concerning overall training strategies.

Flight Crew

The typical flight crew payload operations trainee will possess the following general characteristics:

- Highly educated in engineering/science, Ph.D. or equivalent.
- Technical generalist
- Fluent in English
- Previously demonstrated leadership abilities, or a pattern of growth in responsibility
- Up to date involvement with scientific/engineering developments, preferably hardware oriented. Operational

experience, test experience, independent field work, lab work.

- Culturally and situationally adaptive
- Ability to operate under stress
- Team oriented, good communication skills

For the flight crew, most of the above profile information presents passive rather than active requirements. That is, they imply things which will not have to be done, such as remedial training for language or reading skills, or compensation for attitudinal deficiencies, short attention span, etc. Additionally, methods for courseware presentation will not have to be adjusted for learner limitations, but may be optimized to whatever modality is deemed to be most cost-effective. Due to the rigor of the flight crew selection process, applicants may be expected to be customized for the tasks to be trained, thus allowing a greater degree of flexibility in training methods, modes, and organization.

Overall, the preliminary flight crew profile implies a curriculum which is learner-directed, and learner-paced. Applicants with higher mental aptitude and the capability for independent field work may be expected to take an active role in their learning, supply much of their own motivation and require less positive reinforcement. Instructional presentation can be less structured than is necessary for field dependent, lower aptitude learners. In general, it may be superficially concluded that less flight crew training (and training development) is needed to attain a given proficiency level. The hope, in fact, would be that the flight crew would benefit from a teaching strategy conformed to their aptitudes and abilities.

As a large caveat to the above however, while it is certainly true that learners with higher intrinsic capabilities can acquire skills and knowledges with less external help than others, it is by no means sure that this represents the preferred teaching approach for flight crew applicants. Studies in fact, seeking to match instructional strategies with aptitude or ability groupings have shown that while high aptitude groups perform better under less structured regimen than lower aptitude groups; both groups perform better and experience greater skill retention under more organized approaches. These more organized instructional approaches are characterized by structured step-by-step demonstrations of tasks at a slow presentation rate followed by extensive exercises. Information is presented in simple terms, with small steps and frequent feedback through immediate practice. A functional context is provided for the instructional content, with positive, external reinforcement supplied via

instructor or training device. Implied of course, is that the training device design must allow for these capabilities.

The correct conclusion to be drawn from the trainee profile, therefore, is that while the flight crew trainees should be able to accommodate an unstructured, self-directed curriculum with minimal feedback and intrinsic rather than extrinsic reinforcement; it is by no means an advantage. It may not be as necessary to provide structure for the field independent, higher aptitude learners as it is for the lower aptitude, field dependent workers, but if a clear, logical structure is provided, the flight crew trainees can use and benefit from it. Likewise, small instruction steps and repeated practice may be less necessary for flight crew trainees than for others, but such instructional strategies are advantageous to both high and low ability learners.

Since it is possible (though not certain) that high caliber trainees will react unfavorably to such techniques, the courseware and curriculum used must be flexible enough to permit individual students to proceed at different rates through a training sequence and/or to repeat segments until they are mastered. In the case of CBT courseware for example, learner selected options could be provided to allow branching around auxiliary information, or to proceed immediately to test (for feedback). Simulator training scenarios and supporting materials could be configured in the same manner. This kind of flexibility should serve to speed training and maximize resource effectivity, as well as accommodating individual learner characteristics.

Ground Crew

Profile parameters for the payload operations ground crew are much less defined than those for the flight crew. As far as can be ascertained, there are no current efforts to define desirable characteristics for this class of personnel. Based on Spacelab experience, however, the typical ground operations trainee will probably possess the following general characteristics:

- Well educated, typically a B.S. in engineering or science
- Team oriented, good communication skills, outgoing personality
- Fluency in English.

As with the flight crew, most of the above profile information implies training and techniques which will not have to be accommodated, such as educational remediation. In general though, since the probable personnel requirements and the concomitant screening process used to define eligible applicants will not be

as stringent as that for the flight crew; it can be assumed that the ground crew as a class will exhibit less overall learning abilities than the flight crew. This factor argues favorably for the adoption of structured, high feedback, frequent reinforcement training as described for the flight crews, including the flexibility to allow for individual learner characteristics.

The differences in education, experience and possibly, aptitudes between the two groups suggests that a two-tier course system may be necessary for selected topics (such as experiment familiarization) in which the flight and ground crews must both be trained. This partial duplication of effort can be mitigated by flexibility in the instructional materials (as described in the preceding sections) so that one set of developed courseware can be used by both groups. Experiment familiarization courses for example, which both flight and ground crews might be expected to need, could be geared to the perceived abilities of the ground controllers, but presented in a flexible manner, so that the flight crew could use the same courseware. Even if different facets of the same topic (such as experiment operations) were to be emphasized for each group, the differences could be modularized within each curriculum. Since classroom training does not lend itself to this kind of flexibility, it may be wise to limit the use of classroom training to courses unique to each group. Obviously there are tradeoffs to be considered, such as the cost-effectivity of twin classroom training tracks on the same topic versus flexible self-study materials or CBT courseware.

Summary Recommendations

The overall training strategy for both flight and ground crews should be to provide:

- a) clear and logical structure
- b) small, incremental instruction steps
- c) repeated practice
- d) external feedback and positive reinforcement.

Training devices and instructional materials should be designed to allow small, well structured steps, and a slow presentation rate accompanied by a high rate of repetition. Concepts and instructions should be presented in simple language when possible, and the instructional content should be related in a functional context. External feedback and motivation should be supplied via a live instructor or by features intrinsic to the training device and/or materials.

Training devices and instructional materials should be flexible enough to permit a range of learning rates and the optional repetition of course segments. Specific instruction should be geared to the minimum learning level of the trainees likely to use it, while allowing alternate learning paths for those of greater capability.

The positive aspects of the trainees' group characteristics should not be interpreted as recommending any particular training strategy which seeks to capitalize on group characteristics or specialized aptitudes. Rather the training strategies employed should focus on those techniques found to benefit all aptitude and ability groups. The positive trainee attributes indicated in the profiles contained herein should be used only as a general indicator of the minimum (and not necessarily the optimal) required training and structuring needed by specific groups.

Open Issues/Notes

APPENDIX B

APPENDIX B

PTMS ISSUE OR DESIGN GOAL

Issue Title: Metrics for Student and Training Program Evaluation

Issue No: I-13

Revision: 1

Date: 11/22/89

Assumptions

The Space Station payload training programs will be learner-centered and largely self-paced, utilizing self-reporting as a primary indicator of learning. There will, however, be a need for student performance measurement: to aid students in self-evaluation, to guide their instructors, and to monitor training effectiveness.

There will be an ongoing validation effort throughout the lifetime of each training system to evaluate its effectiveness.

There will be an ongoing validation effort throughout the lifetime of the Space Station to evaluate training development effectiveness.

Discussion and Rationale

Proper determination of evaluation criteria and evaluation mechanisms is important to the success of any training program. These include criteria and mechanisms to examine both student and training program performance (though student performance criteria usually serve as measures of training program efficacy as well). Only the careful selection of appropriate measures for each specific purpose will enable the ultimate capabilities of a training system to be realized.

The properties of the metrics of performance evaluation specified for a training regime can have a major effect on evaluation results, independent of any training benefits. For instance, metrics chosen inappropriately for a course of training can yield results unrelated to actual training objectives. An example of this would be measuring the speed of response to a stimulus such as a radar track, when the accurate analysis of that radar signature is the actual training objective. Other instances of misapplied measures include:

- metrics that are relevant to training objectives but do not yield consistent results
- metrics that are consistent and appropriate but do not respond proportionally to the degree of training.

These and other instances where poorly chosen proficiency measures impact training effectiveness will be examined along with the criteria for accurate measures of task proficiency.

With the advent of modern computer technology, the capability for automated collection and recording of vast numbers of parameters has become almost a given in training situations involving high-fidelity simulators. What is not clearly understood and is, in fact, a chronic problem is the misuse and misunderstanding of the data available. One of the most common mistakes is confusion between physical measures, and behavioral (performance) measures. Simple data recording and reduction are not equivalent to true performance measurement without careful consideration of measure relevance.

Physical measures represent the scaling of physical quantities or events. As such, they have no validity in and of themselves, and do not yet represent behavioral measures. Behavioral measures represent how well an individual performs a specific task and can be derived from physical measures by the systematic addition of training objective and measurement objective information. In other words, meaning is imparted to the measurement set by the addition of training and measurement objectives. Physical measures cannot be behavioral until they are validated as such.

To be validated, a physical measure must first be augmented with a proficiency-related standard and a tolerance. At this point it can be regarded as evaluative information about the system/operator combination. To further validate the metric as a true performance measure for its particular application, various analytical questions should be answered. Examples of these questions are, "Are the factors measured influential in bringing about the desired outcome?" and "Do the measures distinguish between operator and system contributions to total performance?" Once these and other pertinent questions are resolved, the evaluative information becomes a measure of system performance and then of operator performance. This study will explore the various considerations or "tests" that parameters must pass before meaning is attached to them; the study also delineates some of the measurement options for payload training.

Inputs

Issues in Performance Measurement for Military Aviation with Applications to Air Combat Maneuvering; Norman Lane, Essex Corporation, 1986.

Issues in Development, Evaluation and Use of the NASA Preflight Adaptation Trainer (PAT); Robert S. Kennedy, Norman E. Lane, Essex Corporation, 1988.

Conclusions/Solutions

Purposes for Metrics in Training: The first step in proper metric selection is to have a clear understanding of what needs to be measured and for what purpose. This forms the fundamental background against which candidate measures are considered. The intended purpose or proposed usage of the measures helps define the appropriate operations for metrics validation. Most efforts at measurement will address several training purposes at the same time and, thus, must be validated in several ways.

Measurement of task performance on an individual or group basis is done for the following reasons:

- a) To determine the present proficiency or capability of an individual. This could be for many reasons, including qualification for advancement to a later stage of training or feedback to student or instructor about training progress.
- b) To predict an individual's future performance. Usually, this type of measurement is done to increase training program efficiency through early identification and elimination of trainees not likely to succeed in the curriculum. In the PTC application, it is assumed that other screenings on the trainee population have already been performed; thus, this type of measurement will probably find little use in the PTC application.
- c) To diagnose deficiencies and strengths on component processes underlying the skill being acquired. If a student is making unsatisfactory progress or, more likely given the Space Station training environment, if there is a desire to optimize a student's progress, this information will enable concentration on specific problem areas. This situation is anticipated only for the more complex tasks involving event coordination and/or motor skill development.
- d) To determine training effectivity and/or to evaluate alternative training methods. This relates to the collection of group performance evaluation over time.

Classes of Performance Measures: Candidate metrics for evaluation of a specific task can be derived from a variety of sources to measure many aspects of trainee performance. These classes of measures each have strengths and limitations that will tend to recommend or disqualify them for performance evaluation of certain types of Space Station tasks. This study will discuss the most common classes of measures likely to find applicability in payload training as proficiency evaluation criteria.

Criterion-Referenced Measures

One of the most common types of measures or criteria, for performance is the criterion-referenced measure. Implementing this type of performance metric typically involves comparison of system variables to some pre-established objectives and/or standards. These measures, or tests (given the methodology being developed by the PTMS) would be derived from the training objectives developed through task analysis of payload operations. Since it is anticipated that the principal investigators (PIs) will have a major input to the training objectives, the use of this kind of measure would result basically in giving the PI a greater degree of responsibility for training effectiveness. While the use of criteria from such sources practically guarantees the relevance of a measure, there are many other characteristics that must be considered to validate a metric for a particular purpose. These characteristics will be discussed in a later section of this study.

A more general concern is the suitability of criterion tests as a measure for the type of task under consideration. With these measures, "good" performance is equated to doing the job in a prescribed way and demonstrating the capability to meet defined goals or objectives in self-contained task segments. Obviously, care must be taken to ensure that this assumption is valid. For tasks requiring strategic decision making, event coordination, or motor skills or tasks that are reactive in nature and require response to unspecifiable task conditions criterion-referenced measures may not be valid proficiency metrics. Since most payload-related tasks are anticipated to be highly procedural in nature, however, this kind of test should find wide applicability. Also, even for tasks that are not amenable to procedural evaluation, it is likely that adherence to a set of procedural guidelines will be beneficial to the learning process in the early stages of a student's skill acquisition process.

Outcome Measures

Outcome measures are metrics that define task proficiency in the context of the desired terminal behavior of a student. Basically a subset of the criterion-referenced measure, outcome measures concentrate on the top-level behavioral result desired from a course of training. This goal-oriented approach is appealing because of its obvious relevance to training objectives as well as its coverage of all possible performance components necessary for task accomplishment. It does however have three major weaknesses, which preclude its use in some tasks and for some purposes.

The first problem with outcome measurements is that they may fail to detect large proficiency differences between students. This is because different individuals can produce the same outcome using widely varying techniques and procedure orderings that reflect widely divergent skill levels and energy investments. In flight training for example, it is well known that even if two students can accomplish the same maneuver, the more experienced one will accomplish the maneuver more smoothly, more safely, and with greater economy of motion. If an outcome test was used, based on a simple pass/fail criterion, both students would score the same, though one may be grossly inferior to the other. For the same reason, these measures also fail to provide any diagnostic information to aid in remediation, the second problem.

A third problem with outcome measures is that they are sensitive to irrelevant factors that may alter measured results. In other words, the outcome measure gives a final result without consideration for factors outside the scope and control of the training scenario, such as equipment differences or weather. While this is not a major concern for simulator training, where conditions are (or should be) rigidly controlled, if there is a spurious input, the exclusive use of outcome measures will mask it, and deviations in results caused by factors extraneous to the training situation will not be identified as error. It is therefore recommended that these measures be supplemented with other methods more likely to be tolerant of random or systematic effects, such as instructor observation. Given the anticipated PTC training environment, the use of supplementary measures should not prove to be a major problem.

In summary, while much has been said in warning about pitfalls associated with the use of outcome measures, they are often the measure of choice. It is anticipated that they will find application for final evaluations/qualifications for simple tasks and even for complex tasks requiring many skill and knowledge components because of their appealing relevance to task goals. In such uses, however, it is assumed that sufficient analysis has taken place to ensure that the underlying components contributing to task proficiency are well understood.

Process-Related Measures

Many of the problems encountered with criterion- or outcome-related measures may be circumvented through use of metrics, which focus on the underlying task processes or acquisition behaviors, rather than the end goals. For example, as noted in the discussion of outcome measures, student pilots often can perform the same maneuver or flight function with the same results but using a wide range of proficiencies. Measures focusing on the component skills and knowledges underlying the performance of a task will point out major and minor performance

deficiencies, provide information for diagnostics, and reduce or eliminate the effects of factors extraneous to the training situation. Process-related measures should:

- (a) Address the manner (processes) from which an outcome is arrived
- (b) Quantify performance or ability on the task components that account for variance in those outcomes.

This will result in diagnostic measures that indicate a student's performance under other-than-standard task conditions.

It is anticipated that this type of metric will be used to measure proficiency for complex tasks, requiring mixes of different abilities. One method for measuring the proficiency with which tasks of this type are performed is based on overall performance characteristics, such as:

- (a) Economy of effort: less energy and attention required for a given level of performance
- (b) Consistency: constant (desirable) results under varying inputs
- (c) Adaptability: automatic compensation for varying task conditions or reduced feedback
- (d) Procedure and safety: not exceeding procedural or safety limitations while performing the task.

These factors are present to some extent in all skilled tasks.

Obviously, the use of such measures requires a greater understanding of underlying task processes for valid results. Also, since these kinds of measures are more subtle and take the measurement process to a greater level of detail, they will probably require automated parameter-recording facilities. Given a self-paced learning environment and the anticipated caliber and motivation of the ground and flight crews, this type of evaluation may be carried out by the student. Nevertheless, it may still be useful to take objective measurements, to provide feedback to both student and instructor, evaluate the overall training system.

Empirical Measures

Empirical measures of task proficiency are those derived by analysis of training results over time. This approach is oriented toward measuring variables and assigning relative weights to them to compute performance scores. The measures taken and the weighting schemes applied are derived empirically from the

evaluation of factors outside the student-training system, such as sensitivity to student experience levels and changes in task difficulty. Typically, physical measures that can be extracted from the training system are assessed as candidate behavioral measures by linking them empirically to other variables or factors that demonstrate that the physical measure has the required metric characteristics.

The careful analytical derivation of these measures and weighting systems may require prohibitive amounts of data and analysis time to reach conclusions. The Space Station Training Program, with its inherently small numbers of trainees and schedule constraints, might be hard-pressed to supply the resources required to determine empirical measures using rigorous scientific methodologies; however, given the importance of training optimization and training satisfactory to accomplish mission goals, it is probable that much can be done in this arena using more informal methods. Determinations of the necessary processes for achieving a desired task outcome may be done over increments, utilizing student, instructor, and "graduate" comments, as well as rigor in measurement validation, common sense, and on-orbit results.

Subjective Measures

Subjective measures are evaluations made by an informed observer, such as an instructor or by the student himself. Although there is often an attitude among developers of measurement systems that Subjective Measures are "bad" because to the effect of personal bias, while objective measures are "good" for the opposite reason, this is often not the case. When evaluating the performance of tasks involving complex decision making and cognitive skills (such as the monitoring and operation of simultaneous experiments), instructors may be more able to evaluate key components of performance than can hardware/software. Subjective proficiency judgments also can take into account the effects of variances in task conditions, such as student fatigue, equipment variances, etc. Instructors who are themselves proficient in the task to be trained are able to detect the appropriate aspects of performance and evaluate them without the subjectivity introduced in "objective" measurements through decisions about measurement techniques, data-reduction techniques, and data interpretation. The deficiencies of subjective measurement, such as differences of opinion on what constitutes "good" performance, can be overcome by the pooling of judgments across observers and across time.

Composite Measures

Performance of a complex task typically involves many different skills and abilities. Deriving a separate measure for each skill

and then combining the measures according to some weighting system results in a composite score. A composite score, however, may not be a valid indicator of overall task proficiency for the following reasons:

(a) Individuals tend to emphasize different skill components in accomplishing the same task successfully, thus invalidating the weighting system.

(b) Necessary skill components often vary over time as task proficiency increases.

In addition, separate measures of component skills are more meaningful and revealing for diagnostics than any combination of component scores. If a composite score is required, however, one successful method is to ask informed observers to distribute 100 points across a set of measures according to their perceived importance to task acquisition. This weighting system is then used to combine the measured results into a composite score. While this method has been previously used to good effect, it should be cautioned that such scores will have validity only as a measure of overall performance and not of individual proficiencies. As such, they would see application in overall training system evaluation.

Skills and Knowledges for Payload Training: In the study of measures validation for payload training, it is helpful to review the classes of skills and knowledges in which the flight and ground crews must acquire proficiency. Each class will be briefly discussed and cursory recommendations made as to the types of measures appropriate for performance evaluation.

(1) Academic Knowledge

This is the simplest level at which information concerning each experiment, payload operational system, or station system will be imparted to the trainees. The general purpose of this training is to familiarize the students with the processes involved with each system to increase the benefit from later training, which will either provide more in-depth experience to the system or a system to which it relates. This type of training may best be evaluated through criterion-referenced measures or outcome measures (such as the percentage of correct answers) and subjective measures (such as answers to essay questions). These evaluations will probably take the form of written tests or, in the case of CBT, specific electronic queries designed to assess knowledge retention. For the expected caliber of PTC trainees and their anticipated motivational levels, self-report may also be used. The most likely use for metrics at this stage is qualification

for advancement to the next training stage or for student feedback.

(2) Procedural Skills

It is expected that procedural skills will make up the bulk of the training required for payload operations. The use of POIC consoles and experiment and station systems primarily will involve following set procedures and guidelines. This type of training, to the extent that it does not include tasks requiring higher order skills (to be discussed) may be evaluated through criterion- and outcome-referenced measures. It is possible that Subjective Measures, such as instructor or student evaluation, may also find application. Possible uses for metrics at this stage include qualification for advancement, student/instructor feedback, and determinations of training effectivity.

The following three skills/knowledges will be considered to operate together in a multicomponent, heterogenous task.

(3) Perceptual/Interpretive

This kind of skill/knowledge will be required for tasks such as recognizing astronomical or experimental phenomena and interpreting their meaning. While the demonstration of this type of learning may be accomplished through simple pass/fail criteria, it should be noted that this proficiency is seldom exhibited alone and usually works in concert with other high-order abilities to accomplish a higher level task. Thus, this skill will be considered with numbers 4 and 5 below when recommending performance measures.

(4) Cognitive

This refers to decision making based on observed phenomena. As such, it is closely related to Perceptual/Interpretive skills since one generally follows the other in the performance of a task.

(5) Hand/Eye Coordination, Motor Skills

It is expected that the flight crew will utilize these skills for tasks such as installation/removal of experiments, experiment manipulation, and operation of payload support systems.

It is reasonable to assume that in most cases where skills 3, 4, and 5 are used, they will be utilized in concert to accomplish a high-level task. In the conduct of an experiment for example, the crew member could observe an experimental phenomenon, decide what steps to take in reaction to his or her observation, and carry out those steps utilizing hand/eye coordination. These actions

would probably take place in the context of an overall procedural activity. In this case, there is a good possibility that proficient operators will accomplish the overall task in ways different from novices and also from each other, thus reducing the utility of both criterion- and outcome-related measures to assess performance levels.

While the procedural aspects of the overall activity could be evaluated as discussed in (2) above, the addition of the other proficiency components demand that the overall task be evaluated using some combination of empirical, process-related, and/or subjective measures. The purposes for these measures would include qualification, feedback, diagnosis, and determination of training effectivity. While the development and validation of these measures will be significantly more difficult than the effort to develop criterion or outcome measures, it is also true that these types of tasks represent the minority of tasks to be trained for payload operations.

Mental Integration of Separate, Simultaneous Processes

This skill will be required to perform such tasks as monitoring and/or operation of several experiments at the same time. In these cases, there are many ways to perform satisfactorily. It is also true that overall objectives could be satisfied through a performance pattern that would be unsatisfactory for safety, quality or other reasons. Therefore, proper proficiency evaluation should be done via process-related or empirical measures. It is recommended that subjective measures be employed for this skill as well as for (5) above, as a backup confirmation. The purposes for these measures should also follow those of (5) above.

Metrics Characteristics: After the total set of candidate metrics has been determined in the context of their intended purposes, further screening may be done for other metrics criteria that can have a major influence on the degree to which training effectiveness can be demonstrated. The specific purpose for each measure set will determine the metrics criteria used to evaluate it. Diagnostic measures, for example, which are intended to evaluate individual performance deficiencies, would not necessarily be evaluated for completeness, since only specific skill components are of interest. Likewise, proficiency measures intended to evaluate overall task proficiency would probably not have to meet the diagnostic criterion. The criteria used to validate candidate measures for their specific purposes are listed and explained below.

Reliability

The first step in evaluating a candidate measure or measurement set is to determine its reliability. Will repeated measures of a variable, under the same conditions, yield the same results? This concept includes questions of accuracy and precision, which are physical properties, and stability or consistency, which relate to behavior. A failure on either basis will render a measure unreliable to some degree.

There are two main sources of unreliability in the measurement of a variable. These are the phenomena itself and the observation of that phenomena (both subjective and objective). The second source relates to the accuracy and precision of the measurement and, for the case of objective measurements, may be excluded from further discussion. It is assumed that objective measurements will be made accurately (if not correctly), given advances in training technology during the last 20 years. The case for subjective measurements, including their strengths and weaknesses has been discussed in an earlier section.

The first source of unreliability, the phenomenon being measured, can be unreliable because a lack of stability. Stability refers to the property of a measure (phenomenon) to remain stable across time. Some variables, such as blood pressure, are inherently unstable and will vary from trial to trial based on physiological factors beyond the control of the training scenario. Another common example is that of initial student performance. For inexperienced students, skill acquisition is likely to be quite unstable at first, and differences between students are likely to be large. Studies of students performing moderately skilled to highly skilled tasks have shown great differences in size and shape of student skill acquisition curves. Some begin slowly and then progress swiftly, while others learn fast initially and then level off. After performance has stabilized and become asymptotic, proficiency can be reliably measured. Up to that point, measures are more properly indicative of progress and are not reliable predictors of ultimate performance. While these early measures are poor ways to determine an individual's progress, they may have some utility in comparison to normative standards based on previous successful students at an equivalent point in training.

Low reliability can occur for many other reasons. Individual differences among people may be so large that they prevent any generalization of results. Another cause could be measuring the performance of a task that is too easy for the skill level of the group under test. In such a case, the small difference in skill level between individuals could be unimportant compared to other sources of error, thereby giving results unrelated to task performance. Still another cause of unreliability could be using

performance "templates" derived from experts. If the experts are goal oriented, while the students are procedure oriented, measurements of student performance using the criterion of the "expert" templates will be unreliable and possibly irrelevant. The usual result of evaluations utilizing unreliable measures is one of "no differences," since a variable that does not relate to itself in successive measurements cannot be shown to relate to anything else. One obvious implication of this is in the evaluation of alternative training methods using unreliable measures. A "no-differences" conclusion about relative training effectiveness would result in the cheapest method being selected, with no surety that it is indeed the best. Reliability can be considered to be the most fundamental metric characteristic for any measurement purpose. If a measure shows major shifts in time unrelated to training, the effects of training will be difficult or impossible to discern.

Relevance

The relevance of a measure relates to its meaning. A measure is relevant if it accurately reflects the meaning ascribed to it. The measures selected for training evaluation must help determine if the training given has accomplished its purposes. For this to occur, there must be a direct connection between the metric selected and the training offered. There should be a plausible reason why the value of a metric will change in a predetermined direction as a direct result of the intervention.

Attention to relevance will help prevent the establishment of too large a measurement set. The tendency to measure anything that moves can lead to erroneous conclusions based on chance effects. This is especially likely in situations such as flight crew training, where the sample size is small relative to the size of the possible measure set.

The first step in evaluating a measure for relevance is to determine if its content is relevant to its purpose. Presumably, this is done when candidate measures are first determined, based on a training or training program need. Next, the measure must meet a series of empirical requirements:

- (a) Values of the measure must correspond monotonically with the measured skill level. The measure value should increase with practice or time.
- (b) Differences among scores should be primarily related to occurrences of successful events or processes, rather than other factors.
- (c) The measure should show differences between experienced and inexperienced trainees.

(d) A performance measure set should yield quantified information appropriate to its intended purpose.

(e) Values of the measure should match independent estimates from informed sources.

The easiest way of ensuring relevance is to relate the measures as close as possible to the specified training objectives. If taken following training, the measures should be made under conditions and to standards as similar as practical to those obtained during training. If the metric is derived from the training objectives, this should happen automatically and in addition, the metrics derived will be expressed in terms that are observable and, ideally, quantifiable.

On the other hand, while tying the measure to the desired outcome of training ensures relevance, it does not guarantee acceptance by any of the other metrics criteria. (See criterion-referenced measures in a previous section.) In fact, for complex tasks it practically guarantees nonconformance with other essential criteria. In those cases, relevance must be established by linking the candidate measure conceptually to the desired outcome. If empirical data are not available, there must at least be a plausible reason why the candidate measure can be presumed to account for a major part of performance variance.

Sensitivity

Sensitivity reflects the tendency of a measure to change in proportion to the training given. When an individual's capability to perform a task is changed through training, a sensitive measure will show a shift corresponding to the amount or degree of training. An insensitive measure tends to be of limited variability, with most of its variability caused by factors other than those of interest.

One reason for a measure's insensitivity could be the difficulty of the task being measured. If the task is too difficult for the group being measured, it will be insensitive, since a restricted range of scores will result. Similarly, if a task is already highly practiced by the group, it will be difficult to modify through further training and, thus, may not be a sensitive measure of the total training provided. The greatest sensitivity is often exhibited when task difficulty is set so that average performance falls in the midrange of possible scores. This implies that criterion-referenced tests should not be referenced near the top skill level if an accurate spectrum of individual's performances is desired.

Completeness (Dimensionality, Comprehensiveness)

Completeness refers to the degree to which a measure samples the domain of skills and knowledges required for performance. The successful performance of any non-trivial task involves the combination of many task-related skills. A complete measure embodies dimensionality in that it is sensitive to many, if not all, of the relevant aspects of performance. Content of a measure also relates to the extent to which it is sensitive to the relevant factors. Subjective measures have a high potential for completeness, because of the ability of "expert" observers to combine and integrate a set of inherently different measures to arrive at a proficiency evaluation. Observers suitably experienced in the tasks performed, while they may differ on the weights given to all performance components, are probably sensitive to the correct ones, and thus, supply a "complete" measure. The different weightings used necessitate averaging of measures over observers and over time cancel out personal bias.

Separability

Separability refers to the degree with which a measure distinguishes between performance-related student contributions and irrelevant contributions from the training system, the student, and the environment. It is a measure of the tendency for the metric to omit or be insensitive to irrelevant components of performance. Outcome measures, by their nature, often exhibit problems because of their sensitivity to many kinds of factors unrelated to task proficiency. In the case of the student, these might consist of performance instability or momentary shifts in strategy. For the system, these might include equipment variances from such sources as fidelity differentials between trainers, though for the PTC, this is not a serious concern. Environmental factors would include task variables and other uncontrolled aspects of the training scenario. As with training system factors, this is not anticipated to be a big problem for PTC training.

Separability is not as important if the measurement objective is to evaluate each individual's ability to use the system and if each individual does indeed meet the standards. If they do not, however, it becomes important to separate the operator's contributions from contributions caused by irrelevant factors, so that diagnosis may be performed.

Diagnosticity (Specificity)

Diagnostic measures are developed to determine the reasons a particular performance is deficient or proficient. Their specific purpose could be the guidance of a new student or the detection and remediation of a specific difficulty.

Almost any measurement of performance will be comprised of a component related to the student's understanding of a task and a component related to his or her skill in executing that task. Diagnostic metrics must be of sufficient refinement to make distinctions between these two components so that specific and very different remediation for each component can be applied. To be effective in diagnostic use, measures must:

- (a) Provide a level of detail that allows differentiation between skill and knowledge components
- (b) Be sufficiently distinct in the content they measure; that is, metrics that are sensitive to related aspects of performance should not correlate too highly
- (c) Represent each targeted skill by a unique score or combination of scores.

An additional requirement is that diagnostic measures should be applied to individual performances and not to group data. Since the purpose of a diagnostic measure is to evaluate individual deficiencies and since a measure is validated in the context of the purpose it serves, it follows that a diagnostic metric should be used to measure individual and not group differences.

Utility and Cost Benefit (Value against Alternatives)

The utility of a measure refers to its capability to provide accurate results more closely than any other available and affordable measure. This determination involves considerations of both effectiveness of the candidate measure or measurement set against alternative sets and the practicality and feasibility of implementing the measure or measurement system. These considerations are independent of the other metrics criteria discussed above, since they are not resolved by decisions concerning a metric's intrinsic characteristics.

In evaluating the effectiveness of a measure or set of measures against its alternatives, it is necessary to consider the quality of the decisions provided by each measure. This consideration is separate from cost concerns in that two measures that lead to the same decisions are equivalent regardless of cost concerns. Once a set of alternatives has been compared and one is found to produce better results, some judgement will be necessary to determine whether the improvement is worthwhile relative to cost.

In the case of PTC training, it is not expected, at least initially, that cost will be a great issue. PTC training devices will be equipped with instructional features considered standard equipment for high-fidelity simulators, including performance measurement systems. (See Recommendations below.) The selection

of one parameter over another as a performance measure should not entail any additional cost. Later on, as training equipment is upgraded, additional capabilities for the performance measurement function will undoubtedly be considered, at which time, cost will become a factor.

Apart from training effectiveness, the feasibility of a measurement system must also be considered. As an example, student performance measured repeatedly on an experiment simulator is probably a better proficiency measure than a single trial on the flight hardware. It might not be feasible, however, to use the simulator to make a performance evaluation due to late experiment changes. In such a case, utility would direct that the evaluation be carried out on the flight hardware, all other things being equal.

The above example relates to a transient, rather than a steady-state, situation concerning a single experiment. Steady-state issues of system feasibility most often revolve around user acceptance of a given system. Cases abound of instructional features such as performance measurement tools or methodologies that simply are never used. Assuming that the selected measure or set of measures is not useless, the most prevalent reason for user non-acceptance is size and complexity. Modern parameter recording systems are easily capable of generating more data than anyone can assimilate. The designers of the PTC measurement system (the training developers, not the engineers) must ensure that their selected measures do not exceed the instructors' and evaluators' abilities to use them as tools for training and training evaluation.

From the instructor's viewpoint, this means that the feedback from these measures must either reduce workload or increase the instructor's effectiveness. System output must be understandable to the instructor, who should be able to integrate the use of the measure data into the ongoing training flow. Measures that provide summary, or top-level, information are far more likely to be used than large quantities of undigested parameter data. From the training evaluator's viewpoint, this means that, while the data does not have to be "cooked and ready to use," it should be concise, relevant, and manageable in an analytic sense.

RECOMMENDATIONS

Based on the above analysis, the following recommendations may be made concerning the derivation and use of performance measures for Space Station Payload Training:

- (a) Initially, the process of deriving a set of metrics for each experiment should involve justification of a candidate set against clearly established measurement purposes. Next,

the set of measures for each purpose should be validated by evaluation against the metrics criteria described in this study. The metrics criteria, ranked roughly in descending order with respect to their applicability to payload training, are:

- Reliability
- Relevance
- Sensitivity
- Diagnosticity
- Completeness
- Separability
- Utility

Utility is listed last, not because it is least important but because its consideration is independent of the other criteria.

(b) Measurements of student progress in the early stages of skill acquisition should be averaged to reduce the effects of initial skill instabilities and compared against normative curves derived from historical data, rather than directly against other student scores.

(c) Subjective measures of task performance from "expert" observers should be used to verify objective measures of the performance of complex, higher order tasks.

(d) Subjective measures should be used for assessments of overall task performance, rather than component skills.

(e) As proficiency data are collected from PTC operations over time, a systematic validation of measures in use (and their weighting systems) should be performed by correlation with empirical training results.

(f) Quality of instructors is considered to be more important for training effectiveness than sophistication or fidelity of equipment. It is recommended that emphasis be placed on obtaining and/or training skilled instructors as a simple way of boosting training effectiveness (quality and efficiency).

(g) Based on the author's experience, PTC training devices should include the necessary hardware/software to implement an automated performance measurement system with the following general capabilities, under instructor control:

1. Capability to record any of up to 50 software variables available during a training session

2. Capability to plot recorded parameters on strip chart or X-Y plotter as appropriate
3. Capability to provide the above functions, as well as treat the data for use, in real time
4. Capability to present performance results as feedback to the student.

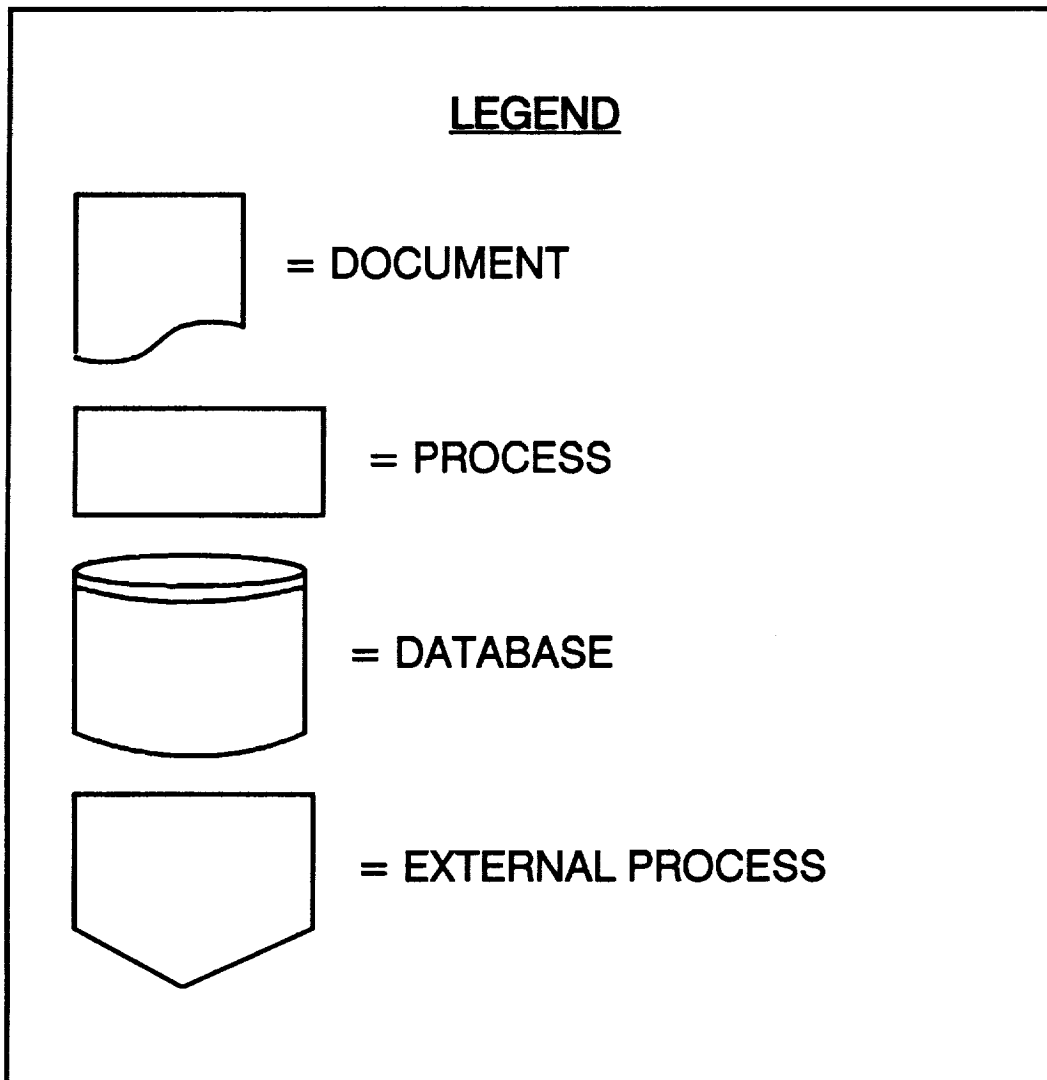
(h) A further study should be commissioned to:

1. Analyze in detail the classes of skills and knowledges necessary for ground/flight payload operations
2. Utilize the characteristics of the identified skills and knowledges to develop a detailed validation procedure for candidate metrics
3. Develop a list of recommended trainer instructional features with respect to automated performance measurement.

OPEN ISSUES/NOTES:

APPENDIX C

APPENDIX C



TRDS Methodology Figures Nomenclature

APPENDIX D

Instructional System Development Tools

Requirements Analysis Tools

Artificial Intelligence Tools

PMTS Survey Report Briefing

Payload Training Methodology Study Survey Report

**Marc Loos
August 24, 1989**

AUTOMATED TOOLS SURVEY

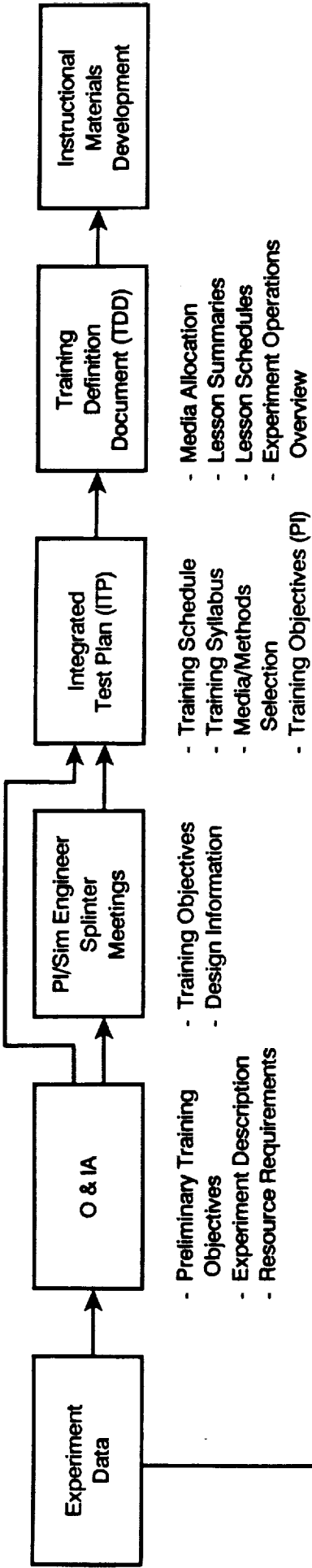
FOR

PTC TRAINING DEVELOPMENT SYSTEM

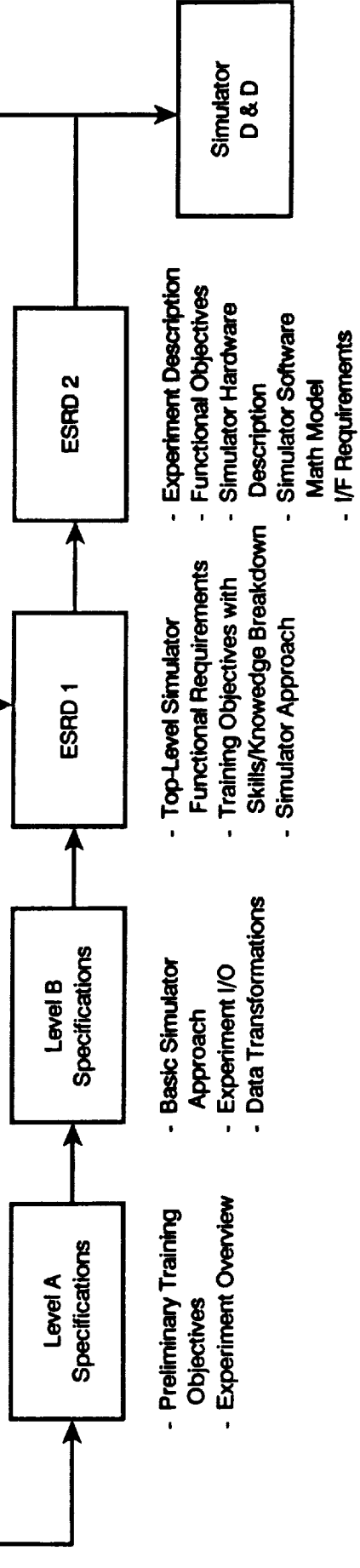
AUTOMATED TOOLS SURVEY FOR PTC TRAINING DEVELOPMENT SYSTEM

- Discussion of PCTC development process and inherent deficiencies
- Discussion of prototype PTC development process
- Survey of tools for automated training analysis (ISD)
- Survey of tools for simulator definition analysis (Systems Engineering)
- Conclusions
- Expert system potential for payload training development

Training Analysis



Simulator Development

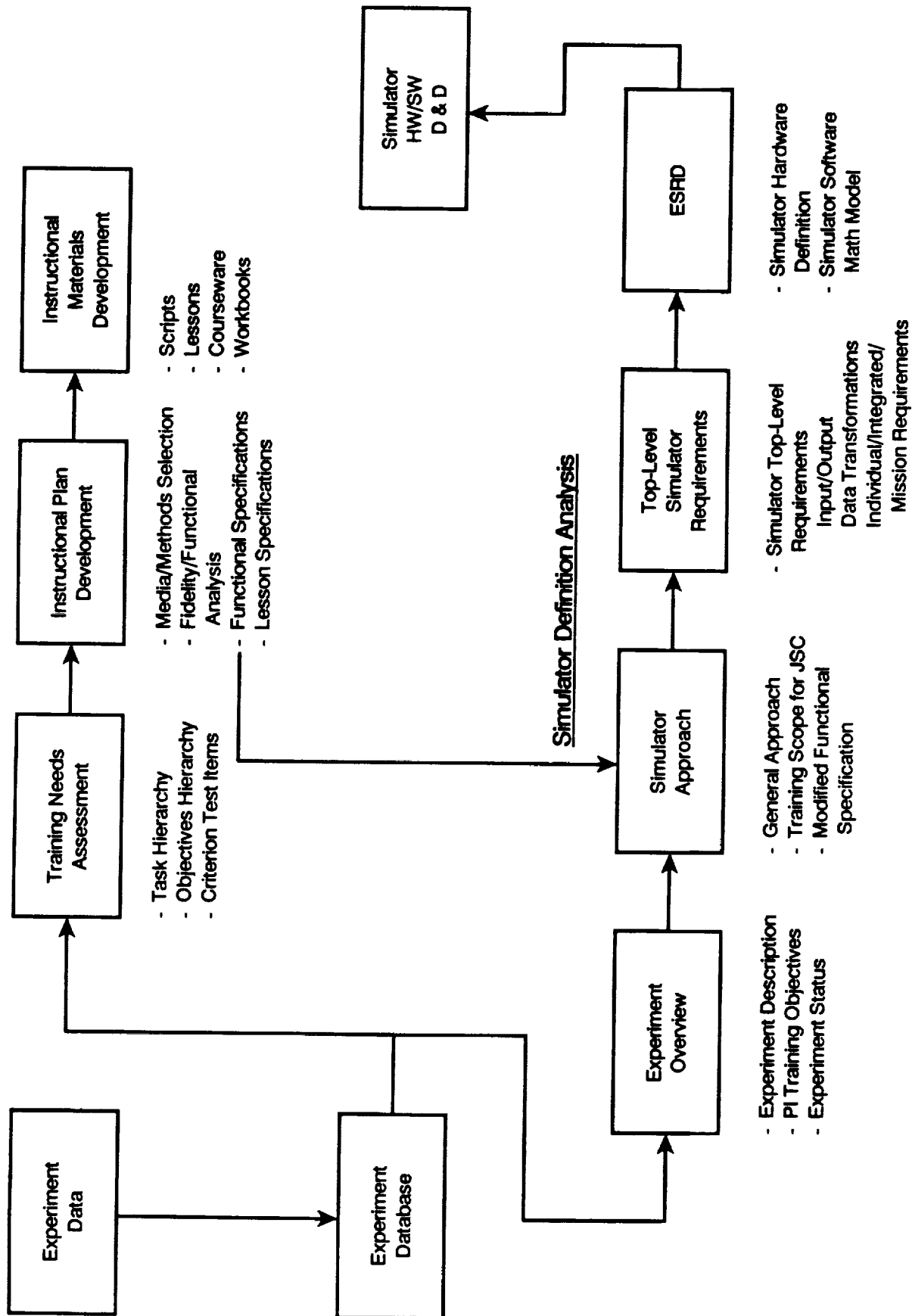


CURRENT PCTC DEVELOPMENT FLOW

CURRENT PCTC TRAINING DEVELOPMENT PROBLEMS

1. No traceability between
 - Experiment information and training objectives
 - Training objectives and simulator requirements
 - Simulator requirements and simulator design
2. No systematic task analysis or development of learning objectives
3. No systematic instructional media and methods selection
4. No fidelity analysis based on training objectives
5. Training analysis data too little, too late for selective, rather than wholesale representation of payload characteristics
6. No mechanisms for organized requirements review and tracking

Training Analysis

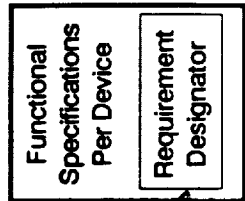
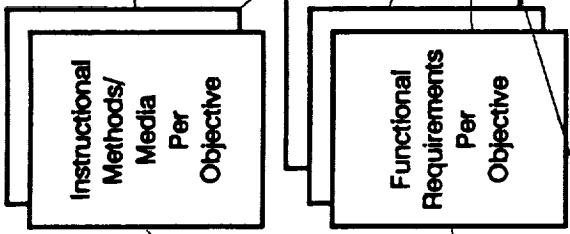


PTC TRAINING DEVELOPMENT FEATURES

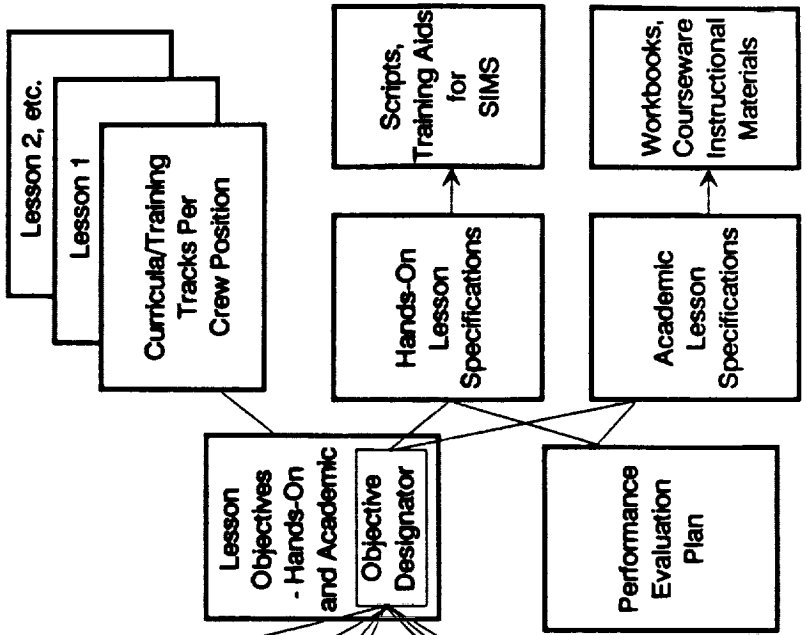
1. End-to-end traceability
experiment data * training tasks * learning objectives * methods/media *
functional requirements * simulator requirements * simulator design
2. Systematic:
 - Task analysis
 - Objectives development
 - Methods/media selection
 - Functional/fidelity analysis
 - Lesson development
3. Cost-effective simulator design based on training objectives and fidelity analyses

INSTRUCTIONAL PLAN DEVELOPMENT

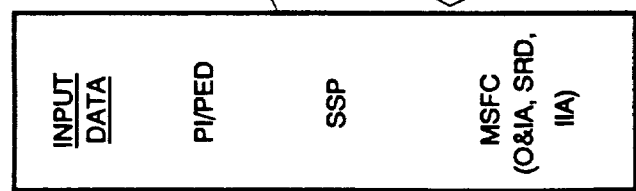
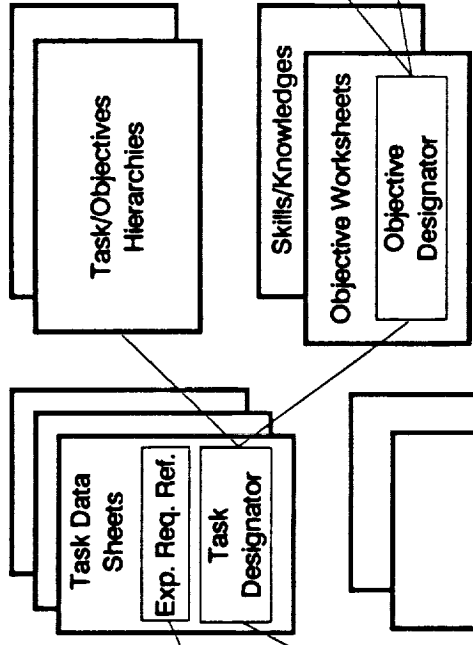
Syllabi Development



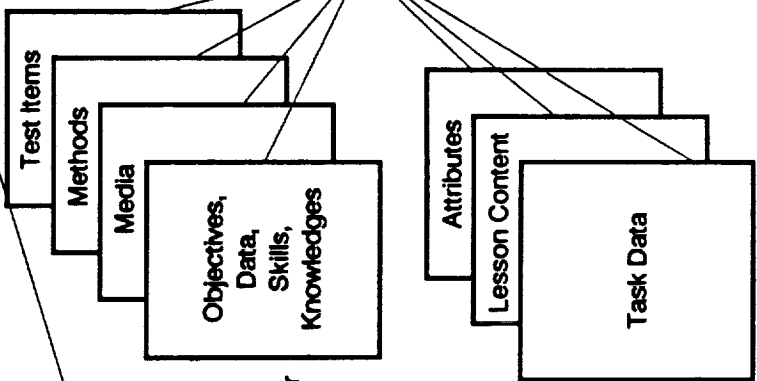
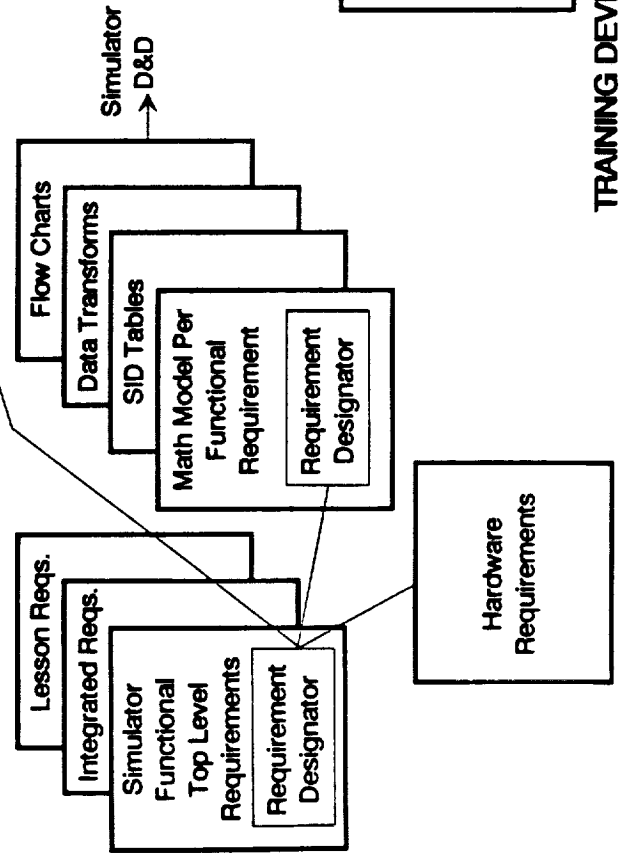
SYLLABI DEVELOPMENT



TRAINING NEEDS ASSESSMENT



SIMULATION DEFINITION ANALYSIS



TRAINING DEVELOPMENT TRACEABILITY

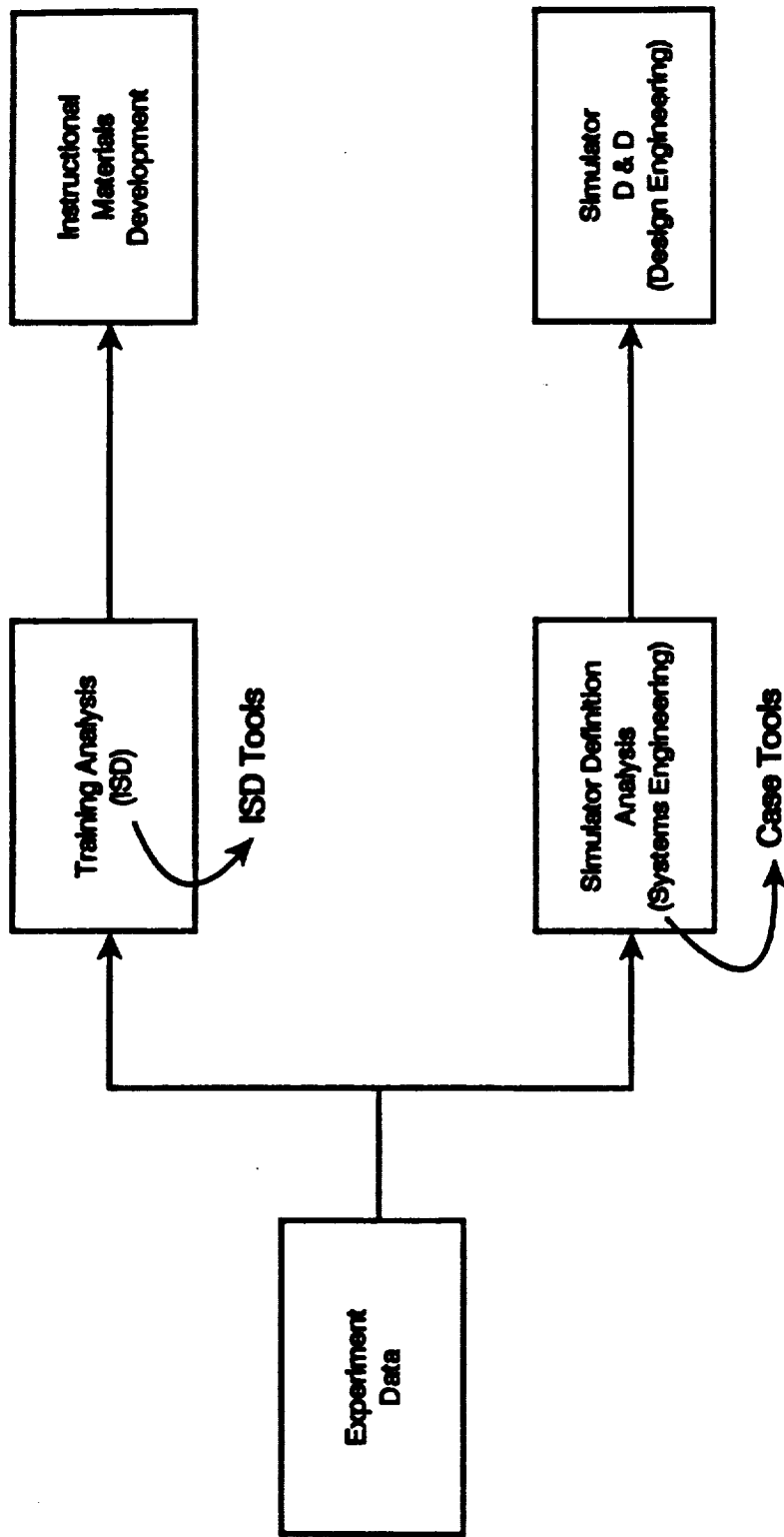
POTENTIAL BENEFITS OF TRAINING DEVELOPMENT AUTOMATION

Why automate training analysis (ISD) ?

- A. Mechanizes a tedious, labor-intensive task
- B. Shortens development time
- C. Can out-perform novice developers in analyses involving many parameters
- D. Faster response to experiment changes
- E. Prompts and guidance for novice developers
- F. Automatic record keeping, reporting
- G. Greater control over process (rigor, consistency)
- H. Automatic traceability

Why automate simulator requirements derivation (Systems Engineering) ?

B., D., E., F., G., H.



■ The training development process should be divided into two distinct areas and addressed separately

- ISD is a specialized requirements derivation process (specialized tools)
- System Engineering is a generalized requirements derivation process performed as a prelude to design and development (CASE tools)
- Distinct and separate software tools address the ISD and System Engineering processes

**ALTERNATIVE STRATEGIES FOR NASA TO ACQUIRE/DEVELOP AN
AUTOMATED ISD CAPABILITY**



A. Develop from scratch with a DBMS

B. Incorporate public-domain or commercially available models into
NASA-developed system

C. Purchase a complete system and modify it to MSFC requirements

D. Contract a vendor to customize their own system to MSFC requirements

E. Contract a training company to analyze problem and design a
system from scratch

COMPANY/TOOL	NASA STRATEGY			
	B	C	D	E
	Integrate Available Models	Purchase a Working System and Customize	Hire Vendor to Customize Own System	Hire Contractor to Develop System from Ground Up
Logicon/ TASCS				
Instructional Science & Dev./ AIDS				
Eagle Technology/ ETRAN				
VEDA, Inc./ CASDAT				
GP Taurio/ ISC				
Courseware, Inc./ CAA				
Barrios Technology/ RDAS				
Allen Corp./ CAD/MTS				
Essex Corp./ ETAS				
Arvin Calspan Corp.				
Advanced Technology				
HUMMRRO				
Booz, Allen, & Hamilton				

Suitability Matrix of ISD Tools for Specific Acquisition Strategies

Note: Only the most appropriate entries for each vendor are indicated

FUNCTIONS/CRITERIA															
VENDOR/TOOL	ISD Functions										Tool Features				
	1	2	3	4	5	6	7	8	9	10*	11*	12*	13*	14	15
Logicon/TASCS															
ISD/AIDS								A							
Eagle Tech./ETRAN								B							
VEDA, Inc./CASDAT															
G.P. Taurio/ISC															
Courseware, Inc./CAA															
Barrios Tech./RDAS								C							
Allen Corp./CAD/MTS								D							
Essex Corp./ETAS															
Nantucket/Clipper								E							

* Automated V & V Enablers

- | | |
|--|--|
| 1 Task analysis | 11 Relational database |
| 2 Objectives development | 12 Reporting capabilities |
| 3 Media selection | 13 Taxonomic task/objective classification |
| 4 Methods selection | 14 Automated task validation |
| 5 Fidelity/functional analysis | 15 Mechanized task data collection |
| 6 Instructional features selection | |
| 7 Lesson/course development | A Task data collection aids |
| 8 Test items/performance evaluation aids | B Interactive user I/F |
| 9 Special features | C Automated courseware authoring |
| 10 Auto-traceability | D Configuration management systems |
| | E Database development systems and database compiler |

Survey of ISD Automation Tools

ALTERNATIVE STRATEGIES FOR NASA TO DEVELOP AN AUTOMATED SIMULATOR

DEFINITION ANALYSIS CAPABILITY



- A. Develop System from Scratch
- B. Modify and integrate commercially available stand-alone tools to mate with software development tool selected by SCS contractor
- C. Customize "comprehensive" case tools

Note: In this case, optimal option selection will be driven by SCS contractor decisions

COMPANY/ TOOL	NASA STRATEGY		
	B	C	
	Integrate Stand Alone Tools with SCS CASE Tool	Customize "Comprehensive" Case Tool	Not Suitable or Not Applicable
NASTEC/ RTrace			
NASTEC/ DesignAid			
Cadre Technologies/ Teamwork			
LBMS/ Automate Plus			
Meta Software/ Meta Design			
Iconix Corp./ Power Tools			
Cadware, Inc./ Sylva Developer Sylva Foundry			
Knowledgeware			
Visible Sys. Corp./ Visible Analyst			
Optima, Inc./ DesignVision			
Helix/ Advantage			
TBE/ RVTS			
TBE/ TAGS			
IDE/ Software Through Pictures			

Suitability Matrix of CASE Tools for Specific Acquisition Strategies

NOTE: Shaded areas indicate available options

VENDOR/TOOL	TOOL CAPABILITIES												
	Strategy B												
	1*	2*	3	4*	5*	6	7	8	9	10	11	12*	
NASTEC/ RTTrace													
Meta Software/ MetaDesign													
TBE/ RVTS													
Nantucket Corp./ Clipper													
	Strategy C												
NASTEC/ Case 2000 DesignAid													
IDE/ Software Through Pictures													
Cadre Technologies/ Teamwork											Soon		
TBE/ TAGS													
Visible Sys. Corp./ Visible Analyst Workbench													
Iconix/ Power Tools													
Knowledgeware													
LBMS/ Auto-Mate Plus													
Cadware/ Sylva Sys. Developer													
Helix/ Advantage													
Optima/ DesignVision													

* Automated V & V Enablers

GENERAL CRITERIA FOR SIMULATOR REQUIREMENTS

SUPPORT TOOLS

- 1 Traceability between levels of simulator requirements
- 2 Traceability from requirements to design elements
- 3 Document generation - (ranges from the ability to port files to external documentation utilities to the capacity for creating interactive document shells)
- 4 Flexible database reporting
- 5 Relational database functionality
- 6 Representation of experiment functional components and their inter-relationships
- 7 Built-in configuration management
- 8 Electronic input capabilities

DESIRABLE CASE TOOL CHARACTERISTICS

- 9 Open architecture
- 10 Real-time modeling
- 11 Robust documentation capabilities or I/Fs to same
- 12 Traceability from requirements to design elements of the CASE tool database

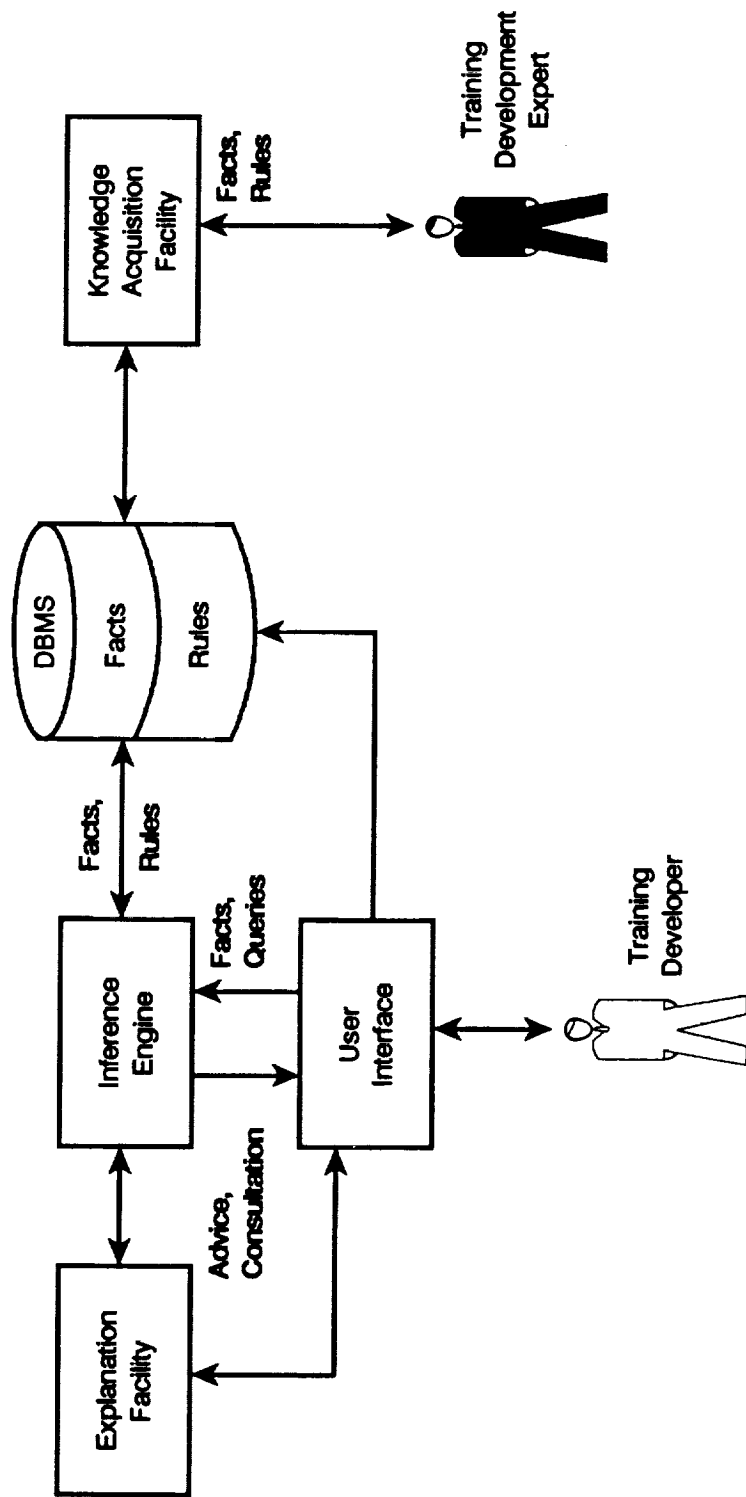
CONCLUSIONS

- The ISD and System Engineering functions of training development represent similar, but distinct processes which may be automated separately
- Comprehensive front-end training analysis functions (ISD) represent a specialized requirements derivation process which is not addressed by CASE tools. Specialized ISD tools exist, however, and can be adapted to meet payload training requirements
- An automated ISD system could feasibly be developed as an outboard system apart from the SCS package
- Simulator requirements derivation, being a Systems Engineering process, is more closely linked to hardware/software design and development and should be considered in that context
- Simulator requirements derivation can be automated with commercially available stand-alone tools and/or "comprehensive" CASE tools
- If a full-featured CASE tool is used for simulator design and development, its capabilities should be utilized to cover the Systems Engineering derivation process as well

**APPLICATION OF EXPERT SYSTEMS
TO DEVELOPMENT
OF PAYLOAD TRAINING**

WHAT IS AN EXPERT SYSTEM ?

- A software program containing a measure of expertise within a limited domain
- A software program that uses knowledge, facts, and reasoning techniques to solve problems normally requiring the ability of human experts
- A software program which can participate in a dialogue with the user to augment the reasoning and inference process of the user
- A software program that can explain its line of reasoning



Rule-Based Expert System

EXPERT SYSTEM COMPONENTS

User Interface - The User Interface is software that allows the User and Expert System to engage in a dialogue. The User can enter situational facts and ask questions, while the Expert System proffers advice and may also ask questions of the user.

Knowledge Acquisition Facility - This facility is software that enables a dialogue between the Expert System and a domain expert or a Knowledge Engineer in order to acquire new knowledge. The facility is also responsible for adding this new knowledge to the Knowledge Base.

Knowledge Base - This is the database which is stored with the facts, rules, and procedures learned from experts. This knowledge is stored in a form which is dictated by the Expert System design.

Explanation Facility - This is software which can relate the chain of reasoning the Expert System followed in arriving at a conclusion.

Inference Engine - The Inference Engine of the Expert System contains the rules and structure which perform the reasoning tasks. The Inference Engine infers new facts from the Knowledge Base and user-provided information by simulating the deductive thought processes of an expert.

WHAT KIND OF PROBLEMS CAN EXPERT SYSTEMS HELP SOLVE?

- Ill-defined, multi-variable problems normally requiring human experience, expertise, and intuition for their solution.
- Problems involving a limited number of choices arrived at by weighing judgement on complicated alternatives
- Complex problems involving information, rules of thumb, and uncertainty
- Problems which can be conceptually defined and solved
- Problems which require cognitive, rather than physical skills

WHAT KIND OF PROBLEMS SHOULD EXPERT SYSTEMS HELP SOLVE?

- Problems with high payoff potential
- Problems in which experts are probably better than amateurs
- Problems where there is general agreement among experts as to the correct solution
- Problems whose solutions can be selected from a hierarchical structure of possible alternatives
- Problems which cannot be solved easily in an algorithmic manner
- Problems which can be solved in a relatively short time, but not instantly

BENEFITS TO SS PAYLOAD TRAINING DEVELOPMENT

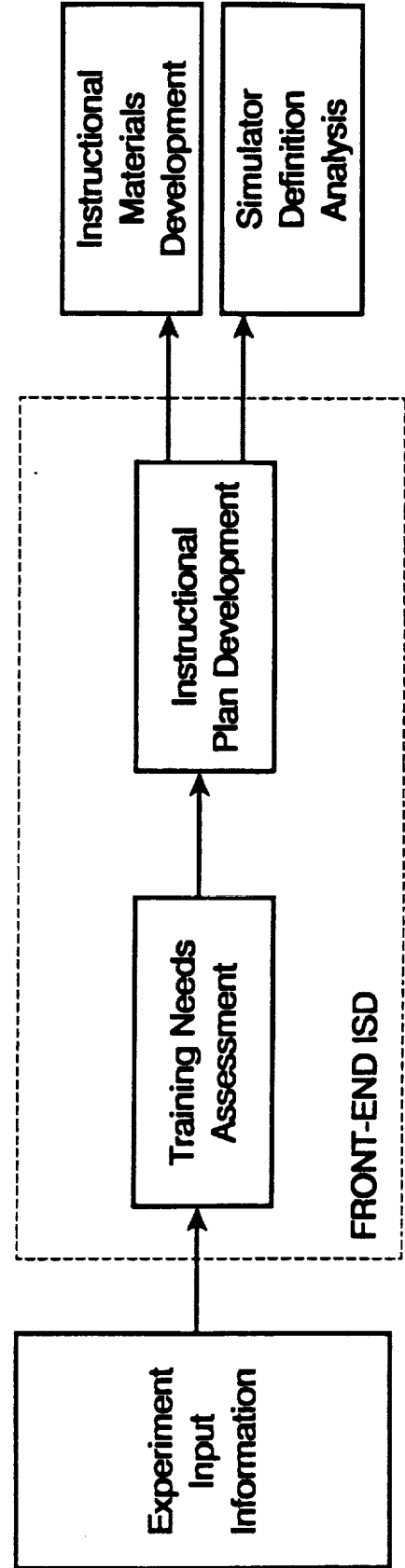
- For the TRDS, an expert system could help novice training developers perform:

TRAINING NEEDS ASSESSMENT

Task Analysis
Task Hierarchy Development
Objectives Development
Test Item Development

INSTRUCTIONAL PLAN DEVELOPMENT

Fidelity/Functional Analysis
Media/Methods Selection
Instructional Features Selection
Lesson Development



BENEFITS TO SS PAYLOAD TRAINING DEVELOPMENT (cont.)

- **An expert system can be easily modified to reflect training development evolution. Subsequent expert system outputs would serve to reinforce the changed philosophy.**

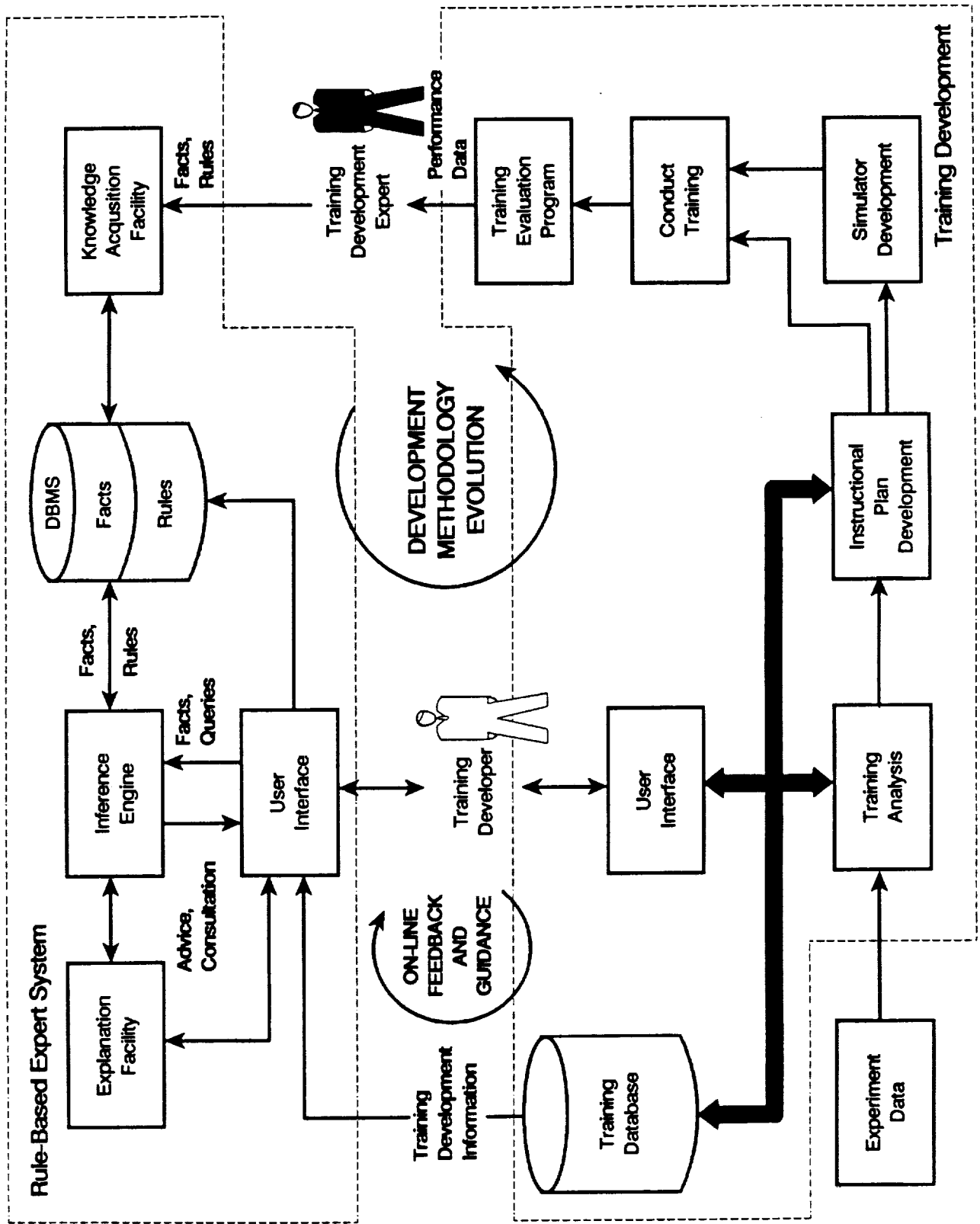
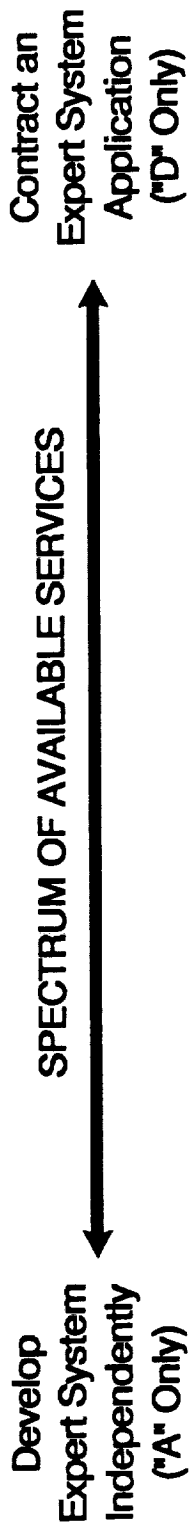


Figure X. Expert System Aiding Training Development Process

HOW IS AN EXPERT SYSTEM DEVELOPED ?

- The process begins with the experiential expert-level knowledge derived by Knowledge Engineers from source material, and from interviews with Subject Matter Experts
- Knowledge Engineers add the derived facts, rules, and procedures which structure an experts reasoning process into Expert System Shells - generic problem solving devices.
- Expert System Shells are software utilities containing an inference mechanism, a skeleton Knowledge Base facility, a User Interface, an Explanation Facility, and a Knowledge Acquisition Facility. With the Expert Shell, the Knowledge Engineer can develop an Expert System for specific problem domains.

HOW CAN AN EXPERT SYSTEM FOR THE TRDS BE DEVELOPED?



D. Develop Specific Expert System Applications

AVAILABLE SERVICES					
COMPANY/TOOL	A	B	C	D	E
	Markets E.S. Shell	E.S. Selection/ Application	Trains Client in Use of E.S. Shell	Develops Expert System Applications	Other
Inference Corp./ Automated Reasoning Tool (Art)	Shaded	Shaded	Shaded	Shaded	
GEMSYM Corp./ G2	Shaded		Shaded	Shaded	
Essex Corp./ Expert System Development		Shaded		Shaded	Shaded
Neuron Data/ Nexpert Object	Shaded		Shaded		
Goldhill Computers, Inc./ Goldworks II	Shaded	Shaded	Shaded	Shaded	
Exsys Inc./ EXSYS	Shaded		Shaded		
Carnegie Group/ Knowledgecraft	Shaded	Shaded	Shaded	Shaded	
Peak Solutions Expert System Development		Shaded		Shaded	Shaded
Logicware International/ MPROLOG					Shaded
Air Force/ Socrates					Shaded

Survey of AI-Related Vendors

NOTE: Shaded areas indicate available options

REFERENCES

1. THE DESIGN AND USE OF EXPERT SYSTEMS/ THE ESSEX CORPORATION
2. A GUIDE TO EXPERT SYSTEMS/ DONALD A. WATERMAN
3. BUILDING EXPERT SYSTEMS/ JAMES MARTIN AND STEVEN OXMAN

**LIST OF SOFTWARE TOOLS FOR AUTOMATION OF TRAINING DEVELOPMENT
FUNCTIONS**

Instructional System Development Tools

Name: Computer-Aided Design and Management of Training
Systems (CAD/MTS)

Vendor: Allen Corporation

Phone: (407) 281-6761

Tool Type: ISD TOOL

Description: CAD/MTS is a comprehensive set of PC-based software tools that automate the complete range of ISD functions. Activities covered include requirements analysis, mission analysis, task analysis, objectives development, and lesson development. The tools were designed to interface with other common PC-based software, such as word processors, spreadsheets, project schedulers, etc. This is intended to aid the user in integrating the ISD applications into existing Analysis and Design procedures. Capabilities of CAD/MTS include:

- (a) Problem/Requirements and Mission Analysis
- (b) Task and Skills Analysis
- (c) Objectives and Objective Hierarchy Development
- (d) Media Selection
- (e) Syllabi Development
- (f) Course Outlines and Scheduling
- (g) Lesson Specifications Development
- (h) Instructor and Student Guides
- (i) Training System Management
- (j) Configuration Management.

Environment: All CAD/MTS applications are designed to run in an IBM PC or compatible environment. All applications utilize a consistent, menu-driven, text-based user interface.

Price: CAD/MTS is a proprietary toolset, generally not for sale. It has been licensed to selective clients who do not threaten Allen's competitive position. The nominal cost of such licensing is generally \$5000.00/copy.

Comments: CAD/MTS can provide traceability from initial requirements to mission requirements, tasks, objectives, and course components. Both built-in and user-defined reports are enabled with a separate report writer application which CAD/MTS was designed to integrate with.

Name: Requirements Definition and Analysis System (RDAS)
Vendor: Lee Wooldridge/Barrios Technology
Phone: (407) 422-2126
Tool Type: ISD TOOL

Description: The Requirements Definition and Analysis System (RDAS) is a software utility designed for use by training analysts to automate the manual procedures involved in task analysis, objectives analysis, methods and media selection, syllabus design, and courseware development. RDAS supports the following training development functions:

(a) Task and Objectives Hierarchy Development - RDAS automates the creation and manipulation of large task and objective databases, limited only by the megabyte storage capabilities of its host. Individual records within the database are located through parent-child relationships which also preserves the hierarchical relationship between the data items. RDAS offers extensive hierarchy rearrangement and editing features, as well as automatic objectives generation from the task hierarchy. RDAS can produce reports and block diagrams for the hierarchies on command.

(b) Job Task Analysis - RDAS aids task analysis through on-line data entry checking, task criticality analysis, and a taxonomic approach to classifying skills and knowledge. RDAS automatically checks for duplicate skills, knowledge, and objectives, and can identify every instance of a data item's use throughout the database. RDAS provides automatic traceability between tasks, objectives, subject-matter sources, skills and knowledge, and all system information.

(c) Methods and Media Selection - RDAS provides an automated methods and media decision aid that can generate alternative sets of either "Hands-On", or "Academic" methods/media recommendations from training objectives data. This includes simulation fidelity requirements suitable for simulator specifications.

(d) Syllabi and Lessons Development - RDAS has facilities for the creation of lessons from the objectives hierarchy, lesson sequencing, creation of courses and curriculum flow from lessons, automatic course and lesson reports.

(e) Custom Capabilities - RDAS contains generic features which can be customized for specific applications. These include automatic courseware

storyboarding, courseware authoring, and test item analysis.

Environment: RDAS is written in dBASE III+/Clipper for use on microcomputers. It can be configured for either single or multiple-user environments.

Price: Licensing arrangements are available.

Comments: RDAS was developed and is owned by Lee Wooldridge who is now working for Barrios Technology. Barrios is an engineering firm currently performing a facility loading study for the SSTF at JSC. In addition, the RDAS Methods and Media Decision Aid is being used to analyze Space Station Crew and Ground Support training requirements. RDAS is available through a direct licensing arrangement with Lee Wooldridge while RDAS support and customization services would be procured through Barrios.

Name: Computer-Aided Analysis (CAA)
Vendor: Courseware, Inc.
Phone: (619) 578-1700
Tool Type: ISD TOOL

Description: Computer-Aided Analysis (CAA) is a software package designed to automate the job/task analysis phase of ISD. It includes a relational database for the storage, organization, and retrieval of task data, and task hierarchies. Based on the input task data, the CAA system automatically selects tasks for training and generates a hierarchy of learning objectives. The training developer then manually edits the hierarchy, adding enabling skills and knowledge to the objectives.

Although the algorithms have been developed for an older, obsolete computer system, CAA does not at present include a methods/media selection and syllabi development facility. CAA supports the following training development functions:

- (a) Task Hierarchy Development - CAA enables the creation of task hierarchies, with task attributes and ancillary information. Tasks may be edited, copied, moved, deleted, and added, with automatic hierarchy reconfiguration. Also, CAA can perform automatic selection of tasks to be trained, if desired.
- (b) Task Validation - A somewhat unique CAA feature is its support for task validation. Upon command, CAA can generate a task validation survey diskette, based upon the tasks derived in the task hierarchy. This diskette is copied, and sent out to Subject Matter Experts who complete the survey and mail the diskettes back. The diskettes are then fed back into CAA which automatically stores the requested information in the task database for validation and other uses. This approach could be generalized to allow the automated collection of task-specific data of all kinds.
- (c) Reports - A variety of reports can be printed, such as task listings, objective listings, hierarchy diagrams, validation reports, and error checks.
- (d) Objective Hierarchy Development - A preliminary objective hierarchy may be automatically generated from the task hierarchy with traceability between tasks and objectives. Most task data is also transferred to what is essentially a new database. CAA then assists the user in completing the hierarchy with requisite skills and knowledge (enabling objectives) learning types, etc. CAA monitors the structure of the task and objective databases for illegal entries.

Environment: CAA is developed to run on the IBM PC or compatible in a stand-alone configuration.

Price: Negotiable

Comments: CAA is primarily a job/task analysis tool. Courseware Inc. has methods/media selection and syllabus development aids as well. These are fielded on an obsolete platform, however, and have not at this date been converted to the PC environment. CAA is a good candidate for customization.

Name: Eagle Training Analysis System (ETRAN)
Vendor: Eagle Technology, Inc.
Phone: (407) 629-6010
Tool Type: ISD Tool

Description: ETRAN is a software package for training analysis consisting of three subsystems: 1) a Relational database management system, 2) a Media Selection System, and 3) an Instructional Features Selection System. Based on the SmartStar relational database management system, the database is used to store all task related data collected about the system to be trained. Once entered, the data can be accessed for modification, sorting, and printing; as well as input to the other ETRAN subsystems.

Media Selection is an algorithmic system which selects the lowest cost media solution to meet the requirements of all the conditions associated with a group of subtasks. For input, it requires the requisite cues, skills, and knowledge for the subtasks, as well as other appropriate task related criteria. These criteria are obtained from coded inputs to the database, as well as from interactive sessions with the training analyst. The Instructional Features Selection System is another algorithmic system which recommends certain options for the selected media, such as a hard disk, or instructor control, which affect the learning environment. As with Media Selection, these choices are based on task information obtained from the database, as well as from interactive sessions with the training analyst.

In general, ETRAN supports the following training development functions:

- (a) Task and Task Hierarchy Development
- (b) Job Task Analysis
- (c) Training Media Selection
- (d) Instructional Features Selection

ETRAN is capable of extensive customization to meet differing requirements. The database format (what is stored and how it is stored) can be modified, as can the format and content of output reports based on training data. The Media Selection System can also be reconfigured to utilize data and conditions specific to each project in making media recommendations.

Environment: ETRAN is currently hosted on a super-mini, connected to remote terminals.

Price: Not costed

Comments: ETRAN is a proprietary system of Eagle Technology. Eagle has expressed interest in MSFC's training requirements and may be willing to discuss customization of their system to fulfill payload training needs.

Name: Essex Training Analysis System (ETAS)
Vendor: Essex Corporation
Phone: (703) 548-4500
Tool Type: ISD TOOL

Description: ETAS is a PC-based software tool, developed to assist the training analyst in the front-end training analysis process. ETAS employs a code table module containing skills, knowledge, references, standards, tools, and equipment to link the various ETAS functions with the ISD process. The ISD functions which ETAS supports include:

(a) **Job/Task Analysis** - ETAS enables the construction of a task structure containing all tasks necessary to operate a system along with task attributes such as task type, activity type, conditions, standards of performance, cues, outputs, safety considerations, etc. Judgments about each task are also entered, such as criticality, frequency and difficulty; skills and knowledge for correct performance. Tasks may be resequenced as appropriate for proper job execution.

(b) **Task Validation** - Task data may be validated by Subject Matter Experts or job incumbents. ETAS generates a Job Performance Measure (checklist) to aid this process. Subject Matter Expert (SME) task data of many kinds may be added to the Task Database in this manner. ETAS can calculate and report descriptive task measures from SME responses.

(c) **Objectives Development** - ETAS aids in the development of objectives hierarchies by allowing the establishment of specific learning objectives, linked to each task. ETAS allows the sequencing of objectives into the order they will be taught to form course outlines. Enabling objectives (skills and knowledge) may be added to each terminal objective. The ETAS Code Table allows the taxonomic storage and retrieval of these skills and knowledge to control learning redundancies.

(d) **Test Item Entry** - ETAS accommodates the addition of Test Items to each task in the database. Test Items can be reviewed and various sorts of tests may be automatically produced by the Test Generator.

Environment: ETAS is designed to run on any IBM PC or compatible PC.

Price:

Comments: ETAS is a systematic approach to training analysis, consisting of three phases. The first phase applies logic to

uncover training requirements, identify causes of problems, and find training solutions. The second phase is task analysis, which creates the data that will serve as the foundation for the training system development. The third phase links tasks to learning objectives.

Name: Instructional Systems Consultant (ISC)
Vendor: G.P. Taurio
Phone: (703) 845-4425
Tool Type: ISD TOOL

Description: ISD is a software package designed to automate the manual, rather than the analytic functions of training development. It includes a relational database management system for the storage, organization, and retrieval of a variety of task related data. The ISC can guide the developer through most of the standard training development functions, including the development of training management and instructional materials, along the guidelines of MIL-STD-1379C. It can help with the development of lesson outlines, training materials specification, and syllabi. It provides the developer with the opportunity to insert methods and media selections, but does not aid them in determining which methods or media to use. It also provides no direct assistance toward determining media functional specifications.

Environment: The ISC is designed to run on IBM PCs which may be networked or not.

Price: \$2000.00/copy.

Comments: This system may be a good candidate for customization to specific MSFC training development requirements. To meet the analytic ISD requirements, it could be enhanced with methods/media selection models and other analytic aids from other sources.

Name: Automated Instructional Design System (AIDS)
Vendor: Instructional Science and Development
Phone: (619) 226-1882
Tool Type: ISD TOOL

Description: AIDS is a utility, intended for use by training analysts to automate the manual procedures involved in job task analysis, training system development, and evaluation. The utility is configured as a series of stand-alone modules integrated under one control system with common database structures. Though flexible and comprehensive, AIDS is designed to provide leverage for the talent and experience of the training analyst, rather than attempting to supplant that experience with analytic algorithms. AIDS supports the following training development functions:

(a) Job Task Analysis - AIDS automates the writing, filing, sorting and printing of tasks, task data, and learning objectives. It employs a taxonomic approach to database building which predefines the verbs, verb-objects and other components of the task statement and task attributes. This enables the automated identification of common tasks, skills and knowledge and ensures standardization in the development of task statements and learning objectives by different developers.

(b) Task Data Collection and Analysis - AIDS can produce survey instruments to assist in data collection from Subject Matter Experts. It is also designed to organize and summarize the survey data in various ways. In addition, ISD is currently developing a utility to enable the assembling of training requirements from input data available on disk.

(c) Objectives Development - AIDS automates the process of assigning Conditions and Standards of Performance to Task Statements in order to generate Learning Objectives. It enables the assembly and modification of Objectives Hierarchies as well as the sequencing of Objectives for learning.

(d) Syllabi and Lessons Development - AIDS assists in defining a Syllabus and assembling Lesson Outlines from AIDS data files. Instructional materials can be developed in an automated fashion with user-defined prompt files and file merge capabilities.

(e) Media and Instructional Features Selection - AIDS employs a model for training media selection which requires the definition of a pool of media/methods, a list of media characteristics required for training, and a measure for how well each media provides the required characteristics. The

model then selects appropriate media candidates for each objective, based on the specific user-defined instructional characteristics and requirements of that objective.

(f) Performance Evaluation - AIDS can generate various worksheets to assist in the evaluation of students, the learning objectives and training system design. Gathered information can then be summarized, analyzed, and documented in a variety of ways.

Environment: AIDS is designed to run on IBM-compatible personal computers. Originally written in BASIC, it is currently being rewritten in C, incorporating Microsoft-Windows tools, and with the ability to make full use of the new operating features of MS-DOS and OS-2. The ability will also be provided to interface with other database management systems such as Lotus and dBase III.

Price: Licensing agreement available for review upon request.

Comments: The media selection capabilities of AIDS has been employed for astronaut in-flight maintenance and Mission Control Center Integrated Communications Officer (INCO) positions. AIDS incorporates flexible document generation capabilities. Allows the user to format and generate documents compiled from the training databases. ISD licenses the software for use, either as a complete package or in self-contained modules. In addition, based on their previous experience/contracts, ISD should be willing to discuss modifying and/or expanding their system to meet MSFC training requirements.

Name: Training Analysis Support Computer System (TASCS)
Vendor: Logicon, Inc.
Phone: (619) 455-1330
Tool Type: ISD TOOL

Description: TASCS is a database oriented tool (developed to interact with dBase IV) which provides a structure for the data derived during training analysis. It provides the means to define and analyze training requirements and to make training system design decisions based on the user's criteria. It also aids in training system revision. The major ISD functions which TASCS supports include:

- (a) Task and Task Hierarchy Development
- (b) Objective and Objectives Hierarchy Development
- (c) Training Media Selection
- (d) Automatic Instructional Method Recommendation
- (e) Lesson Development
- (f) Course Development
- (g) Training Device Definition.

TASCS is designed to be employed iteratively, as available data becomes more in-depth and reliable. TASCS embodies several complex algorithms which use supplied Task and Objectives characteristics to select Methods and Media, calculate required training times, select appropriate testing methods etc.

Environment: PC based; runs under MS DOS, requires 10 MB hard drive.

Price: N/A

Comments: Originally developed for the Air Force MX Program, a dBase III version is available in the public domain. In addition, Logicon would consider selling their latest version, providing training, and/or customizing it for Payload Training development use. Facility with TASCS may be obtained through 2-3 days of orientation.

Name: Computer Aided System for the Design of Aircrew Training
(CASDAT)

Vendor: Veda Inc.

Phone: (407) 658-0044

Tool Type: ISD TOOL

Description: CASDAT is a prototype computer-aided system for the development of aircrew training. It was developed as an experimental model to demonstrate the feasibility of using automation to reduce ISD costs. It supports task list development, objectives hierarchy development, media selection, syllabus development, and lesson specification development.

Environment: CASDAT runs on a PDP-11 in FORTRAN.

Price: N/A

Comments: CASDAT as developed is limited to aircrew training. The underlying methodologies could be used to develop other types of training. Veda is currently working on another automated product designed to fulfill the training development requirements of MIL-STD-1379C.

Requirements Analysis Tools

Name: Teamwork
Vendor: CADRE TECHNOLOGIES
Phone: (407) 425-1528
Tool Type: CASE TOOL

Description: Teamwork is a structured analysis and design tool which can be used to develop functional models, real-time control models, and data models from a single multi-user data base. Teamwork uses the Yourdon-DeMarco methodology for structured analysis and the Constantine-DeMarco methodology for structured design. Teamwork has been developed with industry standard read/write interfaces to allow integrated operations with other software tools for Project Management, Documentation, Configuration Management, etc. Tool features include:

(a) Multi-User Support - the Teamwork data base is designed to allow different team members to share data interactively. Teamwork supports remote network access so that team members can work in different offices, floors, etc.

(b) Documentation - Teamwork has the capability to produce customized documents containing elements of the project database, as well as external components. Document templates for a particular format can be constructed to enable automatic document assembly. In addition, the Teamwork database can interface with outside documentation utilities such as Interleaf, Context, and Scribe. Either method can be used to produce reports, forms, questionnaires, user guides, test plans, and any other program-required documentation.

(c) Configuration Management - Teamwork includes a Configuration Management capability which has provisions for interfacing with other CM tools.

Environment: Teamwork is designed for networked 32 bit workstations, including DEC, Sun, Apollo, HP, and IBM. The Structured Analysis module can also be used with IBM PCs.

Price: Approximately \$9000.00/user for five simultaneous users.

Comments: Teamwork is primarily a requirements analysis tool, typically applied to software development, and therefore is designed to take up where requirements derivation leaves off. It does however, contain relatively versatile documentation capabilities which could be used to assemble any kind of document (for example, an experiment math model), from Teamwork Database and external files in accordance with Document Templates. While Teamwork does not have the capability to build an interactive shell for the automated construction of documents such as an

ESRD, it does have explicit capabilities to interface with specialized utilities which can perform this function. Cadre's basic intent is not to provide desktop publishing capabilities, but rather to allow Teamwork products to be exported to a specialized documentation utility for assembly and editing.

Traceability from the Requirements documents to Teamwork Structured Analysis components at this point, would have to be done with a somewhat manual procedure. In this regard, Cadre is currently working with SAIC to integrate a true requirements traceability utility with Teamwork to provide automatic end-to-end traceability. This enhancement should be available in the fourth quarter of 1989.

Name: Sylva System Developer & Sylva Foundry
Vendor: Cadware, Inc.
Phone: (800) 223-9273
Tool Type: CASE TOOL

Description: System Developer is a structured analysis and design tool which can be customized to support specific methodologies with its RULE TOOL graphical technique definition facility. In addition, it supports almost all major software engineering methodologies. The scope of System Developer extends from software analysis through programming structure chart or pseudocode phases. Interfacing functions allows the user to define intelligent, bi-directional links with other software programs. In addition, the Information Exchange function converts System Developer files to an ASCII neutral format for interfacing with other programs.

Sylva Foundry is a tool which enables the user to build his own CASE tools and his own design techniques from scratch.

Environment: PC workstation-based, under MS-DOS, with individual PC dictionaries, or a team-level dictionary on a LAN with a files-server.

Price: System Developer - \$3000.00; Foundry - \$8500.00

Comments:

Name: Advantage Series
Vendor: Helix Corp.
Phone: (805) 499-0328
Tool Type: CASE TOOL

Description: The Advantage Series is a group of software products, developed primarily for MIS applications. It includes a data dictionary building utility and a system design specification generator. No data-flow or structure chart capability is available. The dictionary builder was designed in Revelation Technology's database management system and can be used for the design of relational databases. It can generate a number of reports based on the contents of the database. The specification generator models screens, reports, and data processes, and generates reports based on the models.

Environment: Advantage runs on IBM or compatible PCs

Price: \$795.00 for each of the two utilities

Comments:

Vendor: Iconix Corp.
Phone: (213) 458-0092
Tool Type: CASE TOOL
Name: PowerTools

Description: PowerTools is a family of five programs for Structured Analysis and Design. They provide structured analysis using the Yourdon-DeMarco methodology, including Data-Flow Diagrams, Mini-Specs and Data Dictionaries. They also support the real-time Ward-Mellor and Hatley methodology. Structured Design is implemented in a top-down, functional decomposition style, using Yourdon-Constantine program structure charts and pseudocode for program design.

Environment: PowerTools are designed to run on the Macintosh line of personal computers. They are compatible with the AppleShare and TOPS LAN systems, and with VAX-based software that emulates Macintosh LANs. PowerTools allows a VAX machine to be used as a file server.

Price: The PowerTools suite costs \$5000.00

Comments: Useful for Software Design and Development.

Name: Software Through Pictures (STP)
Vendor: Interactive Development Environments (IDE)
Phone: (407) 875-5722
Tool Type: CASE TOOL

Description: STP is an integrated environment for software design and development consisting of interlinked software components accessing a shared relational database. The primary tools include a variety of graphics editors which support a large number of popular development methodologies. STP employs an open architecture design, whereby the integration of third-party or proprietary tools is explicitly enabled, as is the customization of the toolset. Prototyping of systems is currently limited to information system models, but STP can be integrated with utilities which can perform engineering prototyping. Major components of the STP environment include:

(a) Graphics Editors - STP contains an integrated family of graphical editors which support several software development methodologies. All editors employ the same user interface, and all allow the user to associate structured information with every object in the diagrams via the Object Annotation Editor. All annotation notes are template-driven.

(b) Automatic Documentation - This application enables the printing of specified subsets of graphic editor diagrams and associated object annotations. In addition, document templates (with built-in prompts) can be designed to enable the interactive creation of specialized documents to fulfill programmatic needs.

(c) Data Dictionary Analysis - This application enables reporting of data dictionary contents according to pre-templates.

(d) DOD-STD-2167 Support - This application includes Object Annotation templates and Document Preparation templates to enable the automated production of 2167-specified data item descriptions (DIDs). Document Templates are user-modifiable. All graphics and tables within the documents are updated automatically from the database.

(e) Document Preparation System - This is a template-driven report generator which can mix text and graphics from the data dictionary and from user inputs. Documents may be output to several popular desktop publishing systems such as Postscript or Interleaf.

(f) Configuration Management - STP supports interfaces to several third party CM systems, as well as the native

version control systems of the various platforms on which it runs.

(g) Requirements Traceability - STP supports traceability internally via traceability reports. It also enables developers to track the results of original customer requirements.

(h) Code Generation - STP can produce the logical outline of an ADA program from the diagrams developed during systems analysis. Once the structure is output, the algorithms and other code can be manually inserted by the system developer.

Environment: STP will run on most of the popular engineering workstations including Apollo, DEC, VAX, HP, and Sun, in a multi-user, simultaneous access mode. STP allows a heterogenous network configuration, employing print servers, file servers, and distributed file systems.

Price: \$21000.00/copy for STP for Real-Time Systems

Comments: STP utilities enable the creation of interactive Document Templates, which could be used to automate the process of writing Simulator Functional Specifications. In addition, interactive templates already exist for MIL-STD-2167 documents which closely resemble the ESRD. Once written, elements of a document can be stored as objects in the database. Since the STP Object Annotation Editor enables the association of references to any database object, sections of documents can be linked to the sections of other documents, and to objects in structured analysis diagrams. Thus, automatic traceability could be established between the Simulator Functional Specification, the ESRD, and the subsequent software development process.

Name: KnowledgeWare
Vendor: KnowledgeWare
Phone: (404) 231-8575
Tool Type: CASE TOOL

Description: Knowledgware is a set of PC-based planning, design, and analysis tools, primarily oriented towards MIS and data processing applications. The approach taken is closer to information engineering, than software engineering. The tools are built around an intelligent Encyclopedia or database, which contains names, definitions, and also their interrelationships.

Environment: Knowledgware tools operate on an IBM Personal System/2 Model 50 or higher or an IBM PC/AT under MS-DOS

Price: \$10000.0 for all three planning, design, and analysis tools

Comments: Utilities seem designed primarily for information systems. This system is not seen as applicable to the development of real-time systems.

Name: Auto-Mate Plus
Vendor: Learmonth & Burchett Mgt. Sys., Inc.
Phone: (800) 231-7515
Tool Type: CASE TOOL

Description: Auto-Mate Plus is a Systems Engineering CASE tool used to develop information systems. It features a data-driven approach, allowing the definition and modeling of data entity structures and their interrelationships. These structures can then be linked into a logical structure. The architecture of an on-line system (including menus) can be constructed and reviewed through a graphics design language.

Environment: PC-based, utilizing a Design Interchange Format, allowing the export of design results to a mainframe for input to other software utilities.

Price: \$8625.00

Comments: This tool seems to be designed with the development of PC-based interactive software utilities in mind; especially those involving extensive database manipulation. As such it could be useful as a tool to create interactive utilities to aid in training development. It does not include any documentation facilities beyond those used to provide information about the interactive application it is building.

Name: MetaDesign
Vendor: Meta Software Corp.
Phone: (617) 576-6920
Tool Type: Graphics Tool

Description: MetaDesign is a diagramming tool for designing and editing complex system models. It can be used to produce flowcharts, presentation graphics, networks, and any other application which involves the depiction of objects or processes, with or without text. All connections made between entities are both graphical and logical. Thus, drawing elements can be manipulated without adversely affecting their interconnections, subordinate objects or text. MetaDesign provides overviews of document hierarchies, and allows easy movement between document levels. Text and graphics can be imported into any MetaDesign document, and MetaDesign diagrams can be exported to other applications.

Environment: MetaDesign runs on IBM family microcomputers based on 80286/386 processors, using the MicroSoft Windows graphics environment.

Price: \$350.00 per instance

Comments: This program could be used to draw a large variety of relational diagrams with integral text. It seems to be capable of generating diagram files which could be merged into documents as needed.

Name: Clipper
Vendor: Nantucket Corp.
Phone: (213) 390-7923
Tool Type: Database Management Tool

Description: Clipper is a dBASE compiler and database development system. It employs an open architecture, which allows the interfacing of external applications such as graphics packages and application generators from third-party vendors. It enables easy menu construction and user-defined functions in Clipper or a variety of other languages. Clipper contains utilities such as a menu-driven debugger, and database file editor to ease the development of database applications.

Environment: Clipper runs on IBM PS\2, PC, AT, XT or 100% compatibles, under MS-DOS.

Name: CASE 2000 DesignAid
Vendor: NASTEC CORP
Phone: (703) 556-9401
Tool Type: CASE TOOL

Description: DesignAid is a structured analysis and design tool which can be used to develop data models, process models, and real-time system models from a single multi-user data base. DesignAid is capable of working with any structured analysis and design technique, including Yourdon/DeMarco, Gane & Sarson, Warnier-Orr, Ward-Mellor, or unique graphic conventions. Tool features include:

(a) Multi-User Support - the DesignAid data base is designed to allow different team members to share data interactively. DesignAid supports remote network access so that team members can work in different offices, floors, etc.

(b) Documentation - DesignAid contains an integrated text and graphics utility, enabling the preparation of customized reports, forms, questionnaires, user guides, test plans, and any other program-required documentation.

Environment: DesignAid can be used on IBM PCs connected with a Local Area Network to a fileserver, or on VAX workstations interfaced with DECnet.

Price:

Comments: DesignAid is primarily a software engineering tool, and therefore is designed to take up where requirements derivation (systems engineering) leaves off. It does, however, contain extensive documentation capabilities which could be used to build any kind of document (such as an experiment math model) and then provide traceability from the document to later analysis components. It does not, however, have the capability to build prompt files which would allow the construction of interactive document shells.

Name: RTrace
Vendor: NASTEC CORP.
Phone: (703) 556-9401
Tool Type: CASE TOOL

Description: RTrace is a Requirements Management database utility which enables the user to load, edit, categorize, and allocate requirements while providing flexible reporting capabilities for these activities. It is designed to work under DoD-STD 2167A, but can be used with any life cycle methodology. RTrace works from electronic documents which can be loaded through Optical Character Recognition, file transfer, etc. It first parses the input documents into individual statements which are then loaded into a multi-user, requirements database. The database may be edited, and requirements added, categorized, or modified as desired. Attributes such as difficulty level, or notes can be assigned to each requirement.

RTrace allows the creation and modification of a system hierarchy (hardware or software) based on the requirements which can define the functional components of a system and their interrelationships. The system hierarchies are revisable and specific requirements can be allocated to each of the system components. In addition, test plans, test cases, and the tests themselves can be linked with individual requirements, as can all files associated with a requirement, such as CAD files, software model files, and documents.

RTrace contains requirements manipulation and documentation facilities to enable the generation of reports covering all aspects of requirements management. These include Requirements Reports, sorted by number, category, or attribute; Requirements Allocation Reports to demonstrate requirements compliance; Traceability Reports, Hierarchy Reports, etc.

Documents based on the developed requirements can be ported back into RTrace in ASCII form to allow the establishment of parent-child relationships between requirements and the more detailed levels of analysis, design, or development.

Environment: RTrace uses an SQL relational data base, running on standard DEC VAX/VMS hardware, and providing full DECnet support for interactive development.

Comments: NASTEC also produces CASE 2000 DESIGNAID

Name: DesignVision
Vendor: Optima, Inc.
Phone: (312) 240-1888
Tool Type: CASE TOOL

Description: DesignVision is a drawing tool developed to support automated systems planning. It is capable of supporting most of the common structured analysis and design methodologies, and allows their customization to or replacement by unique user methods. DesignVision operates under Microsoft Windows which allows the flexibility of accessing other applications such as desktop publishers simultaneously. Traceability can be set up between its structured outputs and resultant code, though it does not have provisions for traceability backward to the design inputs. Documentation capabilities are limited to report generation using database elements, but the Windows application allows interface to other more capable documentation facilities.

Environment: DesignVision runs on IBM-compatible personal computers, supported by Microsoft Windows. The application is presently applied to single users who may access it through a network if desired. In September, the product is slated for a multi-user, concurrent database access configuration through a file server connected to PCs by the Novell LAN.

Price: \$2995.00 per simultaneous user

Name: Technology for the Automated Generation of Systems (TAGS)
Vendor: Teledyne Brown Engineering
Phone: (205) 726-1890
Tool Type: CASE TOOL

Description: TAGS is a computer automated systems and software engineering environment that enables the definition, design, documentation, testing, and maintenance of software/hardware systems. The central concept behind TAGS is that of a graphical system requirements and design language supported by a group of interactive software utilities. These utilities start with the organization of requirements and extend to the generation of Ada code. TAGS contains the following software packages:

(a) Requirements Verification Tool Set (RVTS) - This is actually a stand-alone utility which can build a relational requirements database from input specifications. The specifications (in electronic format) can be scrolled through and identified requirements extracted, labeled, and stored in the requirements database.

The utility supports database editing, query functions, trace matrices, requirements tracing according to user specifications and report capabilities. Direct traceability can be established to the TAGS design database, and the requirements database is accessible to outside applications to enable traceability to other CASE tools.

(b) Storage and Retrieval - This utility implements the automated creation, storage, retrieval, modification, and deletion of system diagrams drawn using the TAGS Input/Output Requirements Language (IORL). The IORL graphics language is said to allow the depiction of every system design aspect including system configuration, inputs and outputs, independent components, interfaces, data types, values, timing constraints, etc. Using this utility, a design relational database composed of hierarchies and groupings of drawings and parameter tables is organized and maintained.

(c) Configuration Management - This menu-driven utility provides electronic forms for problem description, solution description, details of proposed changes, and records of actual changes. It provides support for multiple baselines of the design database (IORL drawing tree), change implementations, change histories, and monitors change status. The CM system does not manage the resultant system Ada code, however, no (legal) code changes can be made without affecting the IORL drawings.

(d) Diagnostic Analyzer - This utility is used to check all IORL documentation for completeness and correctness. This also supposedly guarantees correct syntax for the system Ada code.

(e) Simulation Compiler - This utility can produce a dynamic, discrete event simulation of any section of the IORL structure for early prototyping.

(f) Analysis Library - A variety of functions including various database "look" modes, Database Dictionary, dataflow tracing etc.

(g) Document Processor - Text editor, graphics editor, access interfaces to other documentation facilities such as Postscript, etc.

(h) Automatic Code Generator - Since the IORL methodology accommodates almost all system specifications, Ada code can be directly generated from all or parts of the design database.

Environment: TAGS is designed to run in a distributed workstation environment on Sun, Apollo, VAX, and IBM/RT workstations. RVTS is currently hosted on IBM PC/ATs, but is being modified for workstation use.

Price: Prices per "seat" range from \$6500.00 for a basic system, to \$18000.00 for all the modules. The RVTS is available for \$2250.00/seat.

Comments: The RVTS requirements verification utility can be used separately from the TAGS design environment. With its access features, it would be possible to provide traceability between the requirements database and other CASE tools or other documents such as the ESRD. This would require a specially designed application to tie the desired tools together.

Name: Visible Analyst Workbench
Vendor: Visible Systems Corp.
Phone: (617) 890-2273
Tool Type: CASE TOOL

Description: The Workbench consists of three tools for structured analysis. The first tool, known as the Visible Analyst, is a freeform CAD-like graphics system for data-flow and structure diagrams. The second module, Visible Rules, monitors the diagramming process with either the Yourdon-DeMarco, or Gane and Sarson methodologies. The third tool is called the Visible Dictionary and it is a multi-user, interactive, central data repository for the modeled system. Visible tools operate from a common database which allows simultaneous access by different developers. The Visible Dictionary is available to share information with external databases. In addition, dictionary data can be exported to ASCII files for interchange with other application programs.

Environment: Visible Analyst tools run on the IBM PC, PS/2, 3270. They are configurable for use on LANs running Novell Advanced Netware.

Price: Each Visible Tool sells for \$595.00.

Comments: Outputs from the Visible tools enable code to be developed as the next step, but accommodation for traceability is weak. Visible is currently working on a further enhancement to allow a graphical physical design to be produced prior to code generation. There is no inherent method of tracing design elements back to input documentation.

Artificial Intelligence Tools

Name: Subject Outline Curriculum Resource And Tutoring Expert System (SOCRATES)

Vendor: Air Force

Phone: (205) 293-7031

Tool Type: Expert System

Description: SOCRATES is a rule-based software tool, developed to aid training analysts in the development of lesson outlines. Objective hierarchies comprise the primary system input; lesson plans are its primary output. This process is monitored by SOCRATES which offers advice and guidance according to instructional design rules from recognized leaders in the instructional field (David Merrill and Gagne).

Environment: SOCRATES will run on any IBM PC or compatible. It is comprised of fourteen discs.

Price: Socrates is in the public domain

Comments: SOCRATES is ready for release.

Name: KnowledgeCraft
Vendor: Carnegie Group
Phone: (412) 642-6906
Tool Type: Expert System Shell

Description: KnowledgeCraft is a software toolkit for developing knowledge based systems. It employs Carnegie Representation Language (CRL) which enables a developer to represent all the knowledge pertaining to a problem. Quick prototyping is a tool capability useful for the evaluation of the large systems which KnowledgeCraft seems capable of developing. The tool has an open architecture comprised of eleven modules that may be used separately or together. Though certainly applicable to a wide range of domains, KnowledgeCraft and the Carnegie Group seem predisposed towards manufacturing and continuous processing activities in a production environment.

Environment: KnowledgeCraft runs on DEC and Sun workstations, and symbolic machines such as the TI Explorer and Symbolics.

Comments: Carnegie Group produces and markets its knowledge based products, provides training, and can provide all levels of consultation and application development support.

Name: Expert System Development
Vendor: Essex Corporation
Phone: (703) 548-4500
Tool Type: Expert System

Description: Essex Corporation has an extensive background in Artificial Intelligence. They have developed specific expert system applications as well as standards for evaluating expert systems. They have performed basic research in the areas of expert system design and operation. Essex has participated in the NAVSEA Work Group for Artificial Intelligence for several years, and has formal connections to the Department of Computer Science at Lehigh University and the Advanced Computational Center at the University of Georgia. Essex offers expertise at all levels from basic research to system development across a broad range of application domains.

Comments: Essex can assist in selecting expert system tools for a given application, and can help develop or wholly develop an expert system application for training development.

Name: Exsys
Vendor: Exsys Inc.
Phone: (505) 256-8356
Tool Type: Expert System Shell

Description: Exsys is a relatively inexpensive expert system shell, written in C for greater speed. It includes a rule processing utility to specifically enhance execution speed further. One or two days are required to learn to use Exsys, which contains an automated tutorial. This tool is rule based, with a frame-based extension available. With Exsys, it is possible to read information from external databases, spreadsheets, and equipment.

Environment: Exsys will run on the IBM PC/XT/AT, DEC VAX/VMS, Sun workstations and many UNIX computers.

Price: Starts at \$395.00

Comments: Exsys provides the Exsys tool and also conducts more in-depth training on its use than is provided by the embedded tutorial. They can also provide limited consulting services and can provide referrals to full time knowledge engineering consultants.

Name: G2
Vendor: Gensym Corporation
Phone: (617) 547-9606
Tool Type: Expert System Shell

Description: G2 is a real-time expert system development environment for complex applications requiring continuous and intelligent monitoring, diagnosis, and control. It features a highly sophisticated user interface, employing a windowing system allowing the user to work with knowledge and real time data. Windows can be viewed, hidden, moved, scaled and layered as desired. The interface utilizes interactive graphics and structured natural language, to allow the direct assembling and management of knowledge bases.

G2 contains user customizable interfaces to allow access to sources of real-time data such as control systems data bases and data acquisition equipment.

Environment: G2 is offered on workstations from DEC, Sun, Apple, and HP, and on symbolic computers from TI and Symbolics.

Comments: Gensym provides training for its G2 tool, software customization, interface development, and can develop customized applications.

Name: GoldWorks II
Vendor: Goldhill Computers Inc.
Phone: (617) 621-3300
Tool Type: Expert System Shell

Description: GoldWorks II is a Microsoft Windows application for the development of expert systems. GoldWorks II is both a rules and object oriented tool, written in C and LISP to allow easy extension of the tool to external programs. This tool features a dynamic graphic interface which allows the user to build intelligent graphic interfaces for the resultant application. It employs a menu-driven interface to enable a productive non-programming development environment. Existing databases can be accessed by the application, through a high level dBASE III interface.

Environment: GoldWorks II is compatible with IBM PCs and compatibles, Macintosh, and Sun workstations.

Price: \$8900.00 per unit

Comments: Goldhill is currently involved with expert systems development at MSFC. They produce the tool and also provide all levels of consultation for producing an expert system application.

Name: The Automated Reasoning Tool (ART)
Vendor: Inference Corporation
Phone: (213) 417-7997
Tool Type: Expert System Shell

Description: ART is an expert system shell which includes a development environment and implementation language. ART is a rule based system that is data-directed or driven, so that its internal processing from response to response is determined by the content of the data inputs to it. ART is designed to handle synchronous data input and, due to its speed and response time is capable of functioning in near real time.

Environment: ART will run on most workstations, including VAX, Sun, Apollo, TI, and HP, and on symbolics computers from TI and Symbolics. ART can be installed on a network fileserver. Versions of ART will run on IBM PCs.

Comments: The Inference Corporation produces and markets the ART system. It can provide a wide range of consulting services including training on the ART system and developing specific applications using ART.

Name: MPROLOG
Vendor: Logicware International
Phone: (416) 629-8801
Tool Type: Artificial Intelligence

Description: MPROLOG (Modular PROgramming in LOGic) is a development environment for AI applications in PROLOG. PROLOG is a new, logic-based programming language, which is supposed to surpass the capabilities of most expert system shells.

Environment: Capable of being hosted on IBM, DEC and other environments, including AI workstations such as Sun, Apollo, Macintosh.

Price: \$5.6K - \$17K, depending on host.

Name: Nexpert Object
Vendor: Neuron Data
Phone: (415) 321-4488
Tool Type: Expert System Shell

Description: Nexpert Object is a hybrid rule and objects based shell, written in C and designed to allow embedding into conventional software such as ADA. Nexpert Object can trigger external programs and can directly access popular relational databases. It employs a Microsoft Windows interface for interactive development which can be customized for interactive applications. One strength of the shell is its ability to link disparate databases by mapping fields from multiple records into a consistent object representation. The user has a simple view of database information across several databases.

Environment: Nexpert Object runs on all standard workstations, IBM PCs and compatibles, and Macintosh.

Price: \$5000.00 to \$8000.00, dependent on many factors such as platform used, number of users, etc.

Comments: Neuron Data provides expert system tools and training for their tools. They can provide referrals to applications consultants.

Name: N/A
Vendor: PEAKSolutions
Phone: (612) 854-0228
Tool Type: Artificial Intelligence

Description: PEAKSolutions does not market AI tools, but can produce complete expert systems to order. They have produced an expert system known as Course Builder which captures the techniques and reasoning processes of an acknowledged expert in the education field. This system advises teachers on the best ways to produce curricula for their classes.

Environment: N/A

Price: N/A

Comments: PEAKSolutions is a good example of a company that is not associated with any particular tool and could select the proper tool for a training development expert system. They could develop the expert system or provide various levels of assistance to the effort.

APPENDIX E

APPENDIX E

Issue Title: Training Requirements Development System (TRDS)
Flexibility

Issue No: I-1 Revision: 1

Assumptions

The TRDS must be capable of timely training and/or simulator modification in response to changes in experiments, experiment procedures, or to PI-provided experiment simulators as late as possible in the development cycle.

The TRDS must accommodate continuous updating of training materials and simulators for as long as an experiment is in service.

Discussion and Rationale

Continuous change may be the norm for training and trainers at any point in the development and use of an experiment training system. A change may occur to the experiment itself, or to experiment procedures. Verification or Validation activities could indicate the need for training modifications to support mission objectives. Programmatic concerns may cause shifts in priorities. When a change input is made, the system must be responsive enough to make the necessary adjustments in a timely manner. At the same time, the system must support accurate record keeping. A tight configuration control must be maintained and changes must ripple automatically through all affected supporting documentation.

Conclusions/Solutions

For quick response, all potential sources of changes, updates, modifications, etc., should be considered separately and well-defined procedures installed to deal with each one. The TRDS will provide standardized and preplanned input points for change requests with defined data paths to ensure that all necessary changes to supporting documentation are made. In addition, experiment data and training documentation will be organized in a modular fashion to facilitate easy reshuffling of tasks in response to a shift in job priorities.

Most late changes to the training program for each experiment will be made using a two-track approach. On the first track, to meet the immediate training need, the change will be implemented at as low a level as possible to revise the provided training quickly. Simultaneously on the second track for configuration management, the change will be implemented at the highest

affected level and then flowed down to where the first track was input. At this point an evaluation will be made to verify that the implemented change satisfies all requirements.

Open Issues/Notes

An effective Configuration Management system will be necessary to ensure that the above concerns are satisfied. Since it is probable that the PTC and the TRDS will employ a unified CM system, coordination with the SCS study will be important in this area.

Issue Title: Synergy with SCS Study

Issue No: I-2 Revision: 0

Assumption

The SCS study will specify requirements for a computational facility with the capability to develop, run and maintain software simulations for Space Station payload training.

Discussion and Rationale

The SCS study and the PTMS overlap in responsibilities in several areas. Chief among those is software development. While the PTMS is primarily concerned with front end training analysis, the fruits of this analysis comprise the input to software development which is a function of the SCS. The dividing line between the two is fuzzy at best and may end up being decided by hardware considerations. In addition, it is quite possible that much of the front end activities could best be accomplished using utilities resident on SCS machines. In any case, the requirements analysis and software development efforts must be coordinated to facilitate easy transitioning of simulation data from one activity to the other.

Verification, validation, and configuration management of simulation/simulators comprise other significant overlap areas. The relative roles of the two studies need to be better understood in order to avoid duplication of effort and resources.

Training Results Analysis and the development of instructional aids and overall training strategies are other areas where the PTMS shares responsibility with the SCS study.

Conclusions/Solutions

The purpose of the SCS study is to define top-level functional requirements for the SCS. Based on results thus far, the identified SCS functions cover almost all of the functions presently being analyzed by the PTMS. Therefore, it seems likely that the methodologies and tool recommendations made by the PTMS should be combined in some way with the top-level SCS requirements, before being presented to potential SCS vendors for proposals.

To avoid conflicts between developed SCS requirements and PTMS conclusions, overlap areas between the two studies will be specified and addressed individually. Final system configuration will be determined, pending the outcome of future PTMS trade studies which will examine potential software utilities and target machines to aid implementation of the overlapping

functions. These trades must take into account the needs of the entire development process. SCS requirements and the PTMS recommendations will be coordinated to provide a comprehensive solution.

Open Issues/Notes

Issue Title: Distributed Payload Training and Simulator
Development

Issue No: I-3 Revision: 0

Assumptions

Payload training will start at the PI sites at about L-24 months, using PI provided equipment and training materials. Training will continue at the PTC (starting with classroom and CBT training) at about L-18 months. At L-15, students at the PTC will begin using training materials and training simulators developed by the PED/PI and/or the TRDS. Final integrated training at the SSTF will commence at around L-6, using simulators migrated from the PTC.

Individual payload simulators will be developed both by the TRDS and by the responsible PI/PED. These simulators will be utilized for training in both stand-alone modes and integrated with other simulators which may also have been developed outside of the TRDS.

Discussion and Rationale

Distributed payload training and simulator development create special concerns. One of these involve conflicts between requirements levied by each PI for their individual experiment simulator versus the requirements for a group of simulators integrated together. Others involve the integration of these independently produced simulators into the PTC, as well as integration of training at the PTC with training programs of other centers.

Conclusions/Solutions

In order to deal with requirements conflicts, TRDS methodologies must ensure that a) consideration is given to all requirements, whether based on stand-alone or integrated modes and b) conflicts between the requirements sets are systematically identified and resolved.

Issue I-5 addresses the compatibility problem of externally developed simulators.

While the TRDS is concerned with training development at Marshall, the resultant instruction provided will be only one part of the total curriculum. Integration of PTC-based training with that of other centers will require formal coordination of training programs to avoid duplication of effort and to ensure that overall goals are met. The requirements analysis process will therefore include consideration of program-wide as well as

local training objectives. Formal output to other centers will be prescribed to address the planning, development, scheduling and certification of each component of the total training program. Marshall has already agreed to make developed training materials available to other participating Centers/Agencies. Satisfactory resolution of these intercenter concerns will be a required step in the V&V process.

Open Issues/Notes

Issue Title: Support for Multi-Mission Training Development

Issue No: I-4 **Revision:** 1

Assumptions

Four crew members of an eight man Space Station team will be changed out every 90 days.

From 15% to 25% of the Space Station experiment complement will be changed out each increment.

Training and development are assumed to be accomplished at the PTC on a 40 hours per weekday shift basis with training starting 18 months before launch.

The PTC will be required to support full consolidated increment experiment operations training on one SS increment simultaneously with combined experiment and individual experiment (part-task) training on experiments from three other increments. The consolidated increment trainer set will consist of a U.S. Lab, ESA, JEM, two Nodes, and Attached Payloads trainers. The simulations for these trainers will be interactive. The combined trainer set will be similar to the consolidated increment set except that each trainer will be independent. There will be nine part-task trainers, of which only three will be driven at any one time for training purposes.

The TRDS is assumed to provide all of the training/trainer development needs of the PTC up to a to-be-determined point in the training development cycle. In this sense, the PTC will be the sole direct "customer" for the TRDS.

Discussion and Rationale

The TRDS must be able to produce enough training systems to keep pace with the schedules created by the above conditions. Since PTC capabilities to accommodate payload simulations and different payload increments simultaneously constitute an upper limit to TRDS requirements, this study will coordinate efforts with the SCS study, to obtain an accurate throughput requirement.

Issue Title: Relationship of PI to MSFC Training Organization

Issue No: I-5 Revision: 1

Assumptions

The PI/PED will be responsible for providing whatever information about their experiment or independently developed simulator necessary to develop required training.

The PI/PED will have some role in the training V&V process.

The PTMS is responsible for defining the characteristics of the PI/PED relationship to training development within its defined scope of activities.

Discussion and Rationale

The quantity, quality and timeliness of experiment/simulator data received from the PI/PED is seen as crucial to the efficiency and accuracy of the training development process. The training objectives must be well defined and completely understood for effective training development. Meeting procedures with the PED/PI must be structured for maximum transfer of data and understanding of mission objectives. For his part, the PED/PI needs a clear understanding of the requirements levied upon him by the training function. In particular, the PI needs to understand his responsibilities with regard to individual simulator requirements versus combined simulator requirements.

If the PI/PED supplies a completed simulator, it must be integrated into the overall training plan, curricula must be developed, and its physical interface(s) with the PTC assured by means of a comprehensive Interface Control Document (ICD). Adequate intervals must be allowed for integration and verification of the experiment simulator into the PTC.

Conclusions/Solutions

The PTMS will recommend procedures to maximize the transfer of information to the training developers in a form which will be readily assimilated into the development activity addressed. Methodologies will place an emphasis on defining the type of information and level of detail required from the PED/PI at each stage. Interface documentation will be designed to expedite this transfer of information as well as to ensure common experiment interfaces with the PTC. There may be regularly scheduled meetings between the PI and the training developers to ensure a commonality of goals. Means should be provided to allow the PI oversight into development activities so that problems of interpretation of experiment objectives may be corrected.

Clear agreements should be established with PIs for their support of PTC training in ways such as lectures, simulators, participation in training sessions etc. PI responsibilities for individual simulator requirements and Marshall's responsibilities for combined simulation requirements will be overlapping and designed to ensure compatibility with other simulators. It is recommended that PIs who will be developing their own simulator, be constrained to meet SSE development requirements, select their DEP (if any) from a set of approved alternatives, and that PIs be available to participate in Simulator Validation. Additionally, supplied payload simulators should meet the approved ICD, should include draft operating procedures, and all other necessary flight data file material. ICD composition will be coordinated with the SCS study and other PTC development efforts.

Early access agreements should be arranged for flight/protoflight hardware and software during the development cycle. This would allow exposure to actual hardware for validation of simulator training and identification of differences with the actual experiment.

Rules and guidelines defining PI/PED roles and responsibilities should be available for, and integrated with the start of Phase C procurement activities. For the first launch, experiment procurement activities will begin in late 1989.

Activities and areas where formalized cooperation between the PI/PED and the MSFC training organization is seen as necessary are as follows:

- a) Initial, follow-on interviews
- b) Change/updates to experiments, objectives, procedures etc.
- c) Design of independently-produced simulators (ICD)
- d) Input and participation of PI/PED in V&V and training activities.

Open Issues/Notes

The overall relationship between MSFC and the PI is bounded by policies beyond the scope of this study. These policies must be understood clearly, because they represent constraints on TRDS design.

Issue Title: Continuous Training System Validation Program

Issue No: I-6 **Revision:** 1

Assumptions

There will be an ongoing validation effort throughout the lifetime of each training system to evaluate its effectiveness.

There will be an ongoing validation effort throughout the life of the Space Station to evaluate training development effectiveness.

Discussion and Rationale

The final stage in training development should be a continuous evaluation of the effectiveness and efficiency of the total training curricula in actual use. While the initial training validation process provides immediate feedback, there is also a need for fine tuning of the training system based on actual results. These corrections should be based on trainee performance during training, as well as "in the field".

Conclusions/Issues

The validation methodology will include procedures to solicit and evaluate feedback from trainees and instructors, as well as to compare trainee performance with the performance measurement criteria of the original training objectives. Specific procedures will be defined to feed corrective inputs to the development methodology. Since the SCS study is also considering this function, coordination will be effected between the studies in this area.

In addition, student performance data will be factored with cost and schedule data from the development cycle to evaluate the training development cycle itself. Periodic reviews will be scheduled to identify areas of non-compliance with established performance criteria and plan improvements.

Open Issues/Notes

There is a need to define the appropriate metrics for evaluation of training and training methodology performance.

Issue Title: TRDS Interface with Simulation Software Development and Maintenance Activity

Issue No: I-7 **Revision:** 1

Assumptions

The scope of the PTMS includes all front-end training/trainer development activities up to but not including simulation software development and maintenance.

Software development and maintenance will utilize SSE tools and rules.

Discussion and Rationale

The front-end analysis activities addressed by the PTMS will provide the input requirements to the simulation software development activity. In addition, changes to simulator requirements for any number of reasons will ripple through the development chain to provide corrections and updates to the software maintenance activity. This transfer of information should occur easily and without a significant degree of translation from one process to the other. The requirements development process should output results in a form which the SSE-based tools are designed to accommodate.

Conclusions/Solutions

The current programmatic assumption is that the software development activity will be limited in its methodology to SSE-based tools. Since these tools will define the environment within which the input requirements will be utilized, the PTMS must design a system which will configure its outputs to be directly assimilable by the development activity. The trade studies to be performed as part of the PTMS will include consideration of those software tools endorsed by the SSE. Insofar as there is latitude in the tools which may be utilized within the SSE, this study will monitor the SCS study, which is defining requirements for the SCS vendor, including requirements affecting software development.

Open Issues/Notes

Issue Title: Flexibility for Advanced Technology Insertion

Issue No: I-8 Revision: 0

Assumptions

The Space Station Program life cycle is 30 years, but computers, displays, and other COTS electronic equipment will have to be replaced or upgraded at intervals ranging from 5 to 10 years.

The TRDS will provide all of the training/trainer development needs of the PTC up to a to-be-determined point in the training development cycle. In this sense, the PTC will be the sole direct "customer" for the TRDS.

Discussion and Rationale

The formulation of a durable TRDS must take into account the effects of forthcoming technology developments which have the potential to motivate changes to the initial TRDS configuration. These developments will impinge directly upon the TRDS through the advent of such things as automated training analysis systems. They will also affect the TRDS indirectly by promoting changes to the SCS and the Space Station.

The TRDS must be flexible enough to accommodate emerging technologies and enhanced capabilities. It must be designed in a way which anticipates future trends so that predetermined upgrades may be made easily. This mini-study will predict future trends and extrapolate their effects on the TRDS.

Inputs

SCS Study No. T-20, Onboard Training

SCS Study No. A-6/A-8, Potential For SCS Expansion and Upgrade

PTMS Issue No. I-2, Synergy with SCS Study

Conclusions/Solutions

Future Space Station Trends: While there are many enhancements -- both predetermined and speculative -- planned for the Space Station, most will not have a unique effect on the ways in which training development is conducted. In general, we can expect that many, if not most enhancements, will be motivated by the need to increase Space Station capabilities. More experiments and more complex experiments will be fielded, with a concomitant increase in the amount of training and training development required to support them: therefore, the TRDS must be designed with expansion in mind. This could mean the ability to simply increase the

amount of hardware in the initial system or to rehost system software and data into different, more powerful machines.

Another Space Station development, which at this time remains an open issue is the need for onboard training. This may be required for refresher training or to accommodate unexpected payloads. Other analyses have concluded that this need could be met using (1) books, CBT, and audio/video onboard, (2) on-the-job training with actual payload hardware/software, (3) some type of onboard simulator, or (4) onboard training with responses from a simulator on the ground. Since it is anticipated that books, CBT courseware, payload hardware, and simulators will be developed and utilized for PTC training, there should not be any additional or unique requirements imposed on the TRDS, should onboard training development also be required.

Future SCS Trends: The initial SCS configuration will change over time in response to technological advances, the need for equipment replacement/upgrade, and increases in the training throughput requirement for the PTC. This means that while the SCS functionality may remain constant, the hardware and software to implement it will be steadily improved. Since previous analyses have determined that there is a virtual 100 percent overlap in SCS versus TRDS functions, the impact of these improvements should be felt equally by the TRDS as well.

Technology trends affecting the SCS will be considered for their effects on the TRDS.

Future Trends in ISD and CASE Tools: Tools to aid front-end Systems Engineering and Instructional Systems Development efforts are quickly becoming available or are already available and are undergoing rapid evolution. It is certain that over the lifetime of the Station, upgrades to the original tools recommended will be compelling either because of host hardware upgrades or simply the innate superiority of the newer tools in terms of performance.

Trends in these two areas will be examined in terms of their impact on the TRDS. ISD tools, in particular, hold the promise of future direct applicability to Space Station training needs. It is anticipated that emerging technologies, especially expert systems, will find application as training analysis aids.

Open Issues/Notes

Issue Title: Variation in Knowledge and Proficiency Levels
Required by Different Trainees and Trainee Positions

Issue No: I-9 Revision: 0

Discussion and Rationale

In developing a performance-driven training program for experiment operations, consideration must be given to the differing performance requirements of each position to be trained. In essence, a number of permutations of the same training approach will have to be generated to cover all the needs of each type of crew. In addition, consideration must be given for initial differences in knowledge and skills between trainees. The goal should be to minimize the total amount of training required by providing only the amount of training required for each student to meet established objectives. This consideration will affect not only training development but will also comprise criteria for the associated V&V activities.

Open Issues/Notes

Issue Title: TRDS Ability to Meet Cost/Schedule Commitments

Issue No: I-10 **Revision:** 0

Discussion and Rationale

In addition to satisfying technical and training requirements and reaching design goals, the TRDS must also be sensitive to overall programmatic concerns.

Conclusions/Solutions

The training development process will include continuous feedback to enable timely adjustment of priorities to satisfy cost and schedule commitments. Program review of activities as well as informal meetings with training analysts will occur at appropriate stages in the development cycle to provide this feedback. For example, program-level review of simulator requirements (especially for externally developed simulators) will be a necessary element of the training program flow to ensure that simulator requirements and objectives are met within payload integration cost/schedule constraints.

Open Issues/Notes

There is a need to define metrics by which the progress of training development may be assessed.

Issue Title: Requirements Development Redundancy

Issue No: 11 **Revision:** 0

Assumptions

The scope of the PTMS includes all front-end training/trainer development activities up to but not including simulation software development and maintenance.

Discussion and Rationale

The requirements development process should result in a set of experiment trainer requirements which the software and hardware developers can then use as criteria for and a guide to their design activities. These requirements should not have to be rewritten to satisfy the needs of any particular design effort but should be in a form which can directly progress through increasing levels of detail to a final design. In addition, while the design solution must fit the requirements, the requirements must not specify a particular design solution.

Conclusions/Solutions

To ensure that requirements are only defined once, the simulator designers should become involved with the requirements development process. Their input can assure that the requirements will be expressed in a form that can be directly utilized by design. This involvement with requirements might range from informal reviews of developed materials to performing a structured requirements analysis of simulator functions.

Design team involvement should only begin after the top-level functional and operational simulator requirements have been established. In addition, in-progress verification (where development results are checked against higher level requirements) should be performed by personnel uninvolved with either the design or development efforts. This will preserve the integrity of the process, so that requirements are not made to fit a predetermined solution.

Open Issues/Notes

Issue Title: TRDS Support for Individual, Combined, and Consolidated Increment Training Modes

Issue No: 12 Revision: 0

Assumptions

The TRDS is assumed to provide all of the training/trainer development needs of the PTC up to a to-be-determined point in the training development cycle. In this sense, the PTC will be the sole direct "customer" for the TRDS.

The PTC will be required to support full consolidated increment experiment operations training on one SS increment simultaneously with combined experiment and individual experiment (part-task) training on experiments from three other increments. The consolidated increment trainer set will consist of a U.S. Lab, ESA, JEM, two Nodes, and Attached Payloads trainers. The simulations for these trainers will be interactive. The combined trainer set will be similar to the consolidated increment set except that each trainer will be independent. There will be nine part-task trainers, of which only three will be driven at any one time for training purposes.

Discussion and Rationale

In addition to the development of training for individual payload experiment operations, the TRDS must also account for the requirements levied by combined and consolidated payload training operations. In other words, the methodology should encompass training requirements development flows for each experiment operating individually, interactively with other experiments, and in scenarios dealing with overall mission objectives and constraints. These development flows are usually time-staggered from the individual development schedule.

Conclusions/Solutions

The three development flows to be considered (individual, combined, and consolidated) will follow the same steps and undergo similar review processes. The major differences between them will be the timing of their development steps and the focus of the developed requirements. Their separate concerns will be addressed, starting in the first phase of the TRDS when Job Performance Requirements (JPRs) are compiled. Basically, three overlapping subsets of JPRs and associated attributes will be developed which will each be focused on one of the three training modes. Thus the individual mode subset will consist of tasks to be trained which emphasize experiment activities not requiring other-experiment interaction. Similarly, the combined mode subset will be comprised of all of the individual mode tasks plus those

tasks which require interaction with other experiments in the same module, and so on. These task subsets will be used to separately develop training for each of the three operating modes. Each set of training requirements will follow its own development track and undergo its own review process. Of course the methodology will consider all three in total as well as separately so that unnecessary training redundancies may be avoided and so that a cohesive overall training strategy will result.

The major impact on simulator requirements development will be ensuring (1) that requirements for an individual simulator do not conflict with the requirements of other simulators with which it will interact and (2) that the requirements for each simulator has the necessary capabilities for its intended training application.

The first consideration has been addressed by Issue I-5, "Relationship of PI to MSFC Training Organization," which recommends a clear understanding with the experiment PI concerning individual versus integrated experiment requirements. The second consideration will be met by developing top-level simulator requirements to satisfy the Training Objectives separately derived for each training mode. These Training Objectives will result from the initial division of JPRs discussed above. This will result in three overlapping simulator requirement sets, each customized for a particular training mode. To reduce development effort, it is likely that only one simulator requirements set will be developed for each experiment to meet the needs of all three modes. Thus the same simulator design can be used to meet all simulator training requirements.

Open Issues/Notes

Issue Title: TRDS Evolution Through Feedback Of Empirical Training Results

Issue No: I-15 Revision: 0

Assumptions

There will be an ongoing validation effort throughout the lifetime of the Space Station to evaluate training development effectiveness.

Discussion and Rationale

It is anticipated that, over a 30-year lifetime, the training development system will undergo changes and improvements. Some of these changes will be occasioned by new technology and will affect the development system hardware and software. Other changes will be recommended by training results and will affect the development system methodology.

It is desired to develop a means whereby development methodology effectivity may be determined and systematic improvements made, based on empirical results.

Conclusions/Solutions

Metrics developed for payload training will measure both individual and group task proficiencies. Since development of task proficiency can be regarded as the purpose of the training system, measures of task proficiency may be used to determine training system effectiveness, and by further extension, the effectiveness of the training development methodology. Concepts such as training effectiveness and transfer of training represent the indirect effects, rather than the products of training; therefore, there are no direct measures for them. Rather, these values must be derived or inferred from other training parameters which are measurable, such as task proficiency. To calculate values for effectiveness (either of the training system or the development system), proficiency measures must be combined with other variables, such as the time required for skill acquisition, training development time, training development cost, cost to conduct training, etc. While the PTMS study will provide guidelines, and to some extent, a methodology for applying proficiency measures, the methodologies for calculating effectiveness coefficients range from trivial to complex, depend highly on programmatic imperatives, and are outside the scope of this study.

Once overall effectiveness values for delivered training are determined, however, they may be used to optimize specific facets of the development process. This will be more difficult than

optimizing a specific training system since the development system is one more level of abstraction removed from the results of training. What will be necessary is a direct comparison of training outcomes using two alternative development systems. For example, two methods for determining minimum simulator fidelity levels could be contrasted by comparing training effectiveness values derived from two similar trainers. Each trainer would have to be developed under a methodology differing only in the factor under study. In this way, judgments may be reached concerning alternative training and training development methods.

The preceding discussion implies that, to improve the system, deliberate efforts must be made to collect empirical results and interpret them in accordance with programmatic imperatives (resource utilization). These results are then traced back to their specific causative factors by means of an express testing regime. If a more direct feedback of corrective inputs is desired, then less rigorous methods may be used, with a concomitant loss of certainty and specificity of conclusions. User comments for example, could be collected and intuitively linked with specific development processes, which would then be modified accordingly.

One scenario for this type of corrective procedure would be the following:

Students report that, at the start of combined training, they still feel uncomfortable with certain aspects of a particularly complicated experiment. Rather than employing a rigorous testing regime to scientifically pinpoint the problem, the training administrators modify PTT training to incorporate a "whole-part-whole" concept. With this method, experiment procedures are first demonstrated by the instructor, then practiced in parts by the student, then performed all together. This procedure is used in place of an old PTT method which forced the students to learn every part of the training task at once. When the administrators are satisfied that the students now feel comfortable with the experiment, the development methodology is modified to make the decision for extent of training to be provided more sensitive to experiment complexity.

While the above scenario does not demonstrate a great deal of scientific rigor, it probably represents the manner in which the bulk of methodology improvements will come about. In any case, whether a scientific methodology is used or not to determine optimum development methods, the improvements will not be mechanical and predictable but will each require unique evaluative mechanisms and unique solutions (feedback points), based on the specific situation.

Open Issues/Notes

APPENDIX F

ESSEX CORPORATION

TRAINING REQUIREMENTS DEVELOPMENT SYSTEM (TRDS) FOR
SPACE STATION FREEDOM PAYLOAD TRAINING

ESSEX CORPORATION

The TRDS is a series of methodologies for the systematic development of payload simulators, syllabi, and training programs to teach payload operations. These methodologies include:

1. Training Needs Assessment
2. Instructional Plan Development
3. Simulator Requirements Derivation
4. Training System Verification
5. Training System Validation

ESSEX CORPORATION _____

I. TRDS METHODOLOGIES - INSTRUCTIONAL SYSTEMS DEVELOPMENT (ISD) ACTIVITY

Training Needs Assessment:

Identification of what needs to be taught, in terms of learning objectives and tests

Instructional Plan Development:

Determination of how identified learning objectives will be achieved, and how learning will be measured

ESSEX CORPORATION

II. TRDS METHODOLOGIES - SYSTEMS ENGINEERING ACTIVITY

Simulator Requirements Derivation:

Determination of detailed requirements for simulator design, based on derived training requirements

Training System Verification:

Procedures to ensure the proper implementation of training requirements during the development of simulators, curricula, and instructional materials

Training System Validation:

Procedures to ensure that developed training systems, comprised of simulators, academic materials and instructional aids, fulfill the overall training objectives for which they were designed

ESSEX CORPORATION

TRAINING NEEDS ASSESSMENT

(What must be trained to ensure the successful operation of a specific payload experiment?)

ESSEX CORPORATION _____

TRAINING NEEDS ASSESSMENT

Step One:

Organize experiment and programmatic information into a configuration-controlled database

The Database:

- Gives all training analysts equal access to the same information
- Enables easy identification of data insufficiencies via explicit data organization
- Establishes the basis for traceability from input data to ultimate training system requirements

ESSEX CORPORATION

TRAINING NEEDS ASSESSMENT

Step Two:

Systematically derive a hierarchy of tasks for operating an experiment

F-7

The Task Hierarchy:

- Is comprised of tasks based on, and traceable to experiment data
- Defines tasks and their inter-relationships
- Contains tasks characterized in terms of training-related parameters
- Includes identification of tasks requiring, and not requiring training

ESSEX CORPORATION

Procedure for Developing a Task Hierarchy:

- 1. List the major activities supporting the operations of a particular experiment**
- 2. Select an activity and divide it into phases**
- 3. Walk through each phase, listing all tasks**
- 4. List additional tasks required to perform under extraordinary conditions**
- 5. Specify conditions and standards of performance for each task, as appropriate**
- 6. Assign additional characteristics and attributes as appropriate, to each task**
- 7. Classify each task for training on the basis of its instructional attributes**

ESSEX CORPORATION _____

TRAINING NEEDS ASSESSMENT

Step Three:

Systematically derive a hierarchy of training objectives

The Objectives Hierarchy:

- Is based on and traceable to a task hierarchy
- Defines the trainee behaviors expected after training (what must the training accomplish?)
- Identifies the underlying skills and knowledges necessary to accomplish experiment operational tasks
- Includes the tests used to measure student progress

ESSEX CORPORATION

Procedure for Developing an Objectives Hierarchy:

1. Develop terminal training objectives from the tasks selected for training in the task hierarchy
2. Develop enabling objectives as necessary for appropriate terminal objectives
3. Add higher level objectives to address activities above single experiment operation
4. Develop criterion tests for each terminal objective in an experiment's objective hierarchy
5. Develop diagnostic tests for each enabling objective in an experiment's objective hierarchy

ESSEX CORPORATION _____

INSTRUCTIONAL PLAN DEVELOPMENT

(How will each derived training objective be met for both academic and hands-on instruction?)

ESSEX CORPORATION _____

INSTRUCTIONAL PLAN DEVELOPMENT

Step One:

Determine the instructional methods and media for each training objective

Methods and Media Analysis:

- Initially develops candidate media and methods for each separate objective
- Is based on a determination of learning type for each objective as well as detailed task data from the task hierarchy

ESSEX CORPORATION

Procedures for Media and Methods Selection:

- 1. Identify the learning types associated with each objective**
- 2. Define candidate instructional methods for each objective based on the learning types**
- 3. Screen candidate methods on the basis of student traits, cost, resources, and other external factors**
- 4. Identify hands-on versus academic media objectives**
- 5. Define candidate media for each objective based on:**
 - Compatibility with learning types**
 - General fidelity requirements**
 - Student profiles**
 - Time, student load, cost, resources**

INSTRUCTIONAL PLAN DEVELOPMENT

Step Two:

Develop simulator functional specifications defining the capabilities and features required to train each hands-on training objective

Simulator Requirements Analysis:

- Is based on task data, empirical studies, and lesson specifications
- Results in simulator requirements directly traceable to tasks, lesson goals, or training effectiveness guidelines

Procedures to Determine Simulator Functional Requirements:

1. Derive simulator functional requirements for each hands-on media objective based on:
 - Task analysis data
 - Lesson specifications
 - Empirical studies
2. Organize functional requirements for each objective on simulator functional requirements forms
3. Refine physical and functional fidelity requirements through consideration of the information processing demands of each objective
4. Group objectives into training device categories based on their common and compatible functional requirements
5. Summarize functional requirements into hands-on media functional specifications

ESSEX CORPORATION

INSTRUCTIONAL PLAN DEVELOPMENT

Step Three:

Outline the academic curricula and materials needed to train academic objectives and to support hands-on training

Syllabi Development:

- Produces lesson outlines oriented toward the accomplishment of specific hands-on and academic objectives
- Includes systematic development of scripts and all other materials to support hands-on training
- Establishes training tracks and overall instructional sequencing

ESSEX CORPORATION

Procedures for Syllabi Development:

- 1. Organize and sequence hands-on and academic objectives, tasks, skills, and knowledges into lesson groupings**
- 2. Determine the instructional requirements for each lesson through consideration of:**
 - Numbers and type of personnel**
 - Task criticality, frequency, and difficulty**
 - Marshall and SSF training policies**
- 3. Compose lesson outlines, describing training scenario, methods, media, content, extent of training, feedback, abnormal conditions, instructional aids, alternate learning paths**
- 4. Develop lesson specifications, including performance evaluation plan**

ESSEX CORPORATION _____

SIMULATOR REQUIREMENTS DERIVATION

(What are the detailed simulator design requirements necessary to train the derived hands-on training objectives?)

SIMULATOR REQUIREMENTS DERIVATION

Step one:

Prepare an experiment overview report (EOR)

Plan EOR:

- Describes the experiment in a manner designed to aid simulator requirements development
- Identifies all available experiment data
- Documents PI-provided training objectives
- Documents what is required from other experiments

ESSEX CORPORATION

Procedure for EOR Assembly:

- 1. Report on the current status of experiment development; give a future prognosis of the development effort.**
- 2. Compile integrated experiment simulator requirements**
- 3. List the pi-provided training objectives**
- 4. Produce a summary outline description of available experiment information**

SIMULATOR REQUIREMENTS DERIVATION

Step Two:

Produce a finalized simulator functional specification

A Simulator Functional Specification:

- Describes the simulator capabilities necessary to train a defined set of objectives
- Is based on and traceable to lesson specifications, the EOR, and the initial specification set for each experiment
- Describes the training scope of each simulator
- Is sensitive to PI-provided training objectives, high level (non task-specific) training objectives, PTC resources, and the needs of other simulators

ESSEX CORPORATION

Procedure for Finalizing the Simulator Functional Specification:

- 1. Revise hands-on media functional specifications to incorporate:**
 - . Integrated experiment simulator requirements**
 - . Lesson specifications**
 - . High level training objectives**
 - . PI-provided objectives**

- 2. Record rationales for all functional specification revisions. Report appropriate changes to syllabi development.**

- 3. Compile a list of training objectives for each experiment for JSC coordination with JSC SSF training program**

ESSEX CORPORATION _____

SIMULATOR REQUIREMENTS DERIVATION METHODOLOGY

Step Three:

Prepare a hardware/software Simulator Approach Document (SAD)

A SAD:

- Defines the combination of simulator components necessary to satisfy the simulator functional specification
- Is based on and traceable to the simulator functional specifications, hands-on training objectives, the status of experiment development, and PTC training policies
- Provides a documented initial game plan for all later training development activities
- Provides an early heads-up for training resource allocation planning

ESSEX CORPORATION

Procedure for Producing a SAD:

- 1. Consider the level of PI involvement in PTC training**
- 2. Consider the availability of experiment development hardware or software**
- 3. Consider experiment design factors that indicate the suitability of experiment components for training (simulation vs stimulation)**
- 4. Consider hardware vs software simulation issues**
- 5. Validate final simulator approach with media functional specification. Record rationale for decisions.**
- 6. Provide inputs for increment training plans**
- 7. Develop preliminary ptc trainer allocations as an input to increment training plans**

ESSEX CORPORATION

SIMULATOR REQUIREMENTS DERIVATION METHODOLOGY

Step Four:

Prepare a Top-Level Simulator Requirements Document (TLSRD)

A TLSRD:

- Provides an integrated description of all simulator components, their top-level implementations, required fidelities, and the specific requirements which each component satisfies
- Maps the simulator functional capabilities from the simulator specifications onto appropriate simulator components as defined in the SAD.
- Identifies experiment data insufficiencies
- Is designed to be assembled largely from pre-existing materials such as the EOR, SAD, and the simulator functional specifications

ESSEX CORPORATION

Procedure for Producing a TLRSD:

1. Integrate elements of the EOR, SAD, Simulator Functional Specifications, and Training Objectives Hierarchy
2. Develop simulator element descriptions from the elements described in the SAD
3. Map simulator functional requirements to the appropriate simulator element
4. Assemble simulator interface requirements
5. Identify deficiencies in available experiment data
6. Structure a PI interview to correct deficiencies

ESSEX CORPORATION

SIMULATOR REQUIREMENTS DERIVATION METHODOLOGY

Step Five:

Prepare a detailed Experiment Simulator Requirements Document (ESRD)

An ESRD:

- Describes hardware/software implementation requirements in sufficient detail to enable simulator design and development
- Is based on and traceable to initial experiment data through the TLRSD, simulator functional specifications, and the simulator training objectives
- Includes interface requirements, modeling requirements, and hardware mockup requirements

ESSEX CORPORATION

Procedure for Producing an ESRD:

- 1. Derive detailed interface requirements for each simulator element**
- 2. Derive detailed modeling requirements for each simulator element**
- 3. Derive detailed hardware mockup requirements for each simulator element**

ESSEX CORPORATION _____

TRAINING DEVELOPMENT VERIFICATION METHODOLOGY

(Are the experiment training requirements being properly implemented during training development?)

TRAINING DEVELOPMENT VERIFICATION METHODOLOGY

- Checks the products of each development phase against the requirements established during the previous phase:
 - Training Objectives
 - Instructional Plan
 - Simulator Requirements
 - Simulator Design
 - Simulator Code

TRAINING DEVELOPMENT VERIFICATION METHODOLOGY

- Incorporates three levels of verification activity:
 - Increment independent planning (master verification and test plan)
 - Specification verification (in-progress verification)
 - Verification testing (conducting and planning software/hardware tests)

- Includes responsibilities, deliverables and methods for each phase of verification

GENERAL SEQUENCE OF VERIFICATION ACTIVITIES

1. Produce a generic master verification and test plan as a verification guide for the development of all training systems
2. Perform specification verification at each stage of the development process. The five stages of specification verification are:
 - a. Training objectives verification
 - b. Instructional plan verification
 - c. Simulator requirements verification
 - d. Design verification
 - e. Implementation verification
3. Plan and conduct verification testing.
4. Report verification results at the Simulation Acceptance Review (SAR)

ESSEX CORPORATION _____

TRAINING SYSTEM VALIDATION

(Does the total training system developed for each experiment fulfill the overall training objectives?)

ESSEX CORPORATION

TRAINING SYSTEM VALIDATION METHODOLOGY

- Analyzes the integrated use of all training system components and personnel
- Compares training systems with their basis training objectives and functional requirements
- Is performed in a realistic environment with respect to the ultimate training situation
- Considers all stages and types of training in assessing system adequacy
- Includes procedures for ongoing validation as well as the validation of academic and hands-on training
- Outlines validation test plans in terms of purpose, format, content, and validation methods

ESSEX CORPORATION

GENERAL SEQUENCE OF VALIDATION ACTIVITIES

Initial Validation

1. Formulate test plans for validation of academic instruction
2. Perform academic validation testing
3. Report validation results
4. Formulate test plans for validation of hands-on instruction
5. Perform hands-on validation testing
6. Report validation results

ESSEX CORPORATION

GENERAL SEQUENCE OF VALIDATION ACTIVITIES (continued)

Ongoing Validation of Training Systems in Use

7. Derive performance measures which can be used to evaluate training effectiveness and diagnose training problems
8. Evaluate student performance during training according to derived performance measures
9. Modify training or the training methodology based on performance evaluation