AL-CR-1991-0001

AD-A238 766

DTIC
ELECTE
JUL 0 9 1991
S C D

TRAINING DECISIONS MODELING

SIMULATION ENVIRONMENT:

DESIGN PLAN (CDRL #5);

HARDWARE OPTIONS AND RECOMMENDATIONS (CDRL #8)

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

Submitted to

1LT. Keric Chin
Air Force Human Resources Laboratory
Training Systems Division
Brooks Air Force Base, Texas 78235

Submitted by

Robert J. Glushko
Search Technology
4725 Peachtree Corners Circle
Norcross, Georgia 30092

Alphatech Subcontract No. 086084-3240, Task 89-0010

March 15, 1990

91-04413

91 7 8 1000

NOTICES

This report is published as received and has not been edited by the technical editing staff of the Armstrong Laboratory.

Publication of this report does not constitute approval or disapproval of the ideas or findings. It is published in the interest of scientific and technical information (STINFO) exchange.

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

The Office of Public Affairs has reviewed this report, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This report has been reviewed and is approved for publication.

KERIC B. O. CHIN, Captain, USAF
Contract Monitor

HENDRICK W. RUCK, Technical Director
Technical Training Research Division

RODGER D. BALLENTINE, Colonel, USAF
Chief, Technical Training Research Division

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
May 1991

3. REPORT TYPE AND DATES COVERED
Final Report– November 1989–March 1991

4. TITLE AND SUBTITLE
Training Decisions Modeling Simulation Environment: Design Plan and Hardware Options and Recommendations

5. FUNDING NUMBERS
C - F33615-86-C-0545
PE - 62205F
PR - 1121
TA - 12
WU - 04

6. AUTHOR(S)
Robert J. Glushko

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Alphatech, Incorporated
2 Burlington Executive Center
111 Middelesex Turnpike
Burlington, MA 01803

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)
Armstrong Laboratory
Human Resources Directorate
Technical Training Research Division
Brooks Air Force Base, TX 78235-5000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AL-CR-1991-0001

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This document summarizes the design plan for a Training Decisions Modeling Simulation Environment. The simulation environment is intended to demonstrate the emerging capabilities of the Training Decisions System (TDS) under development at the Air Force Human Resources Laboratory. The design plan specifies the TDS capabilities which will be incorporated into the simulation environment and a plan for the design of the environment itself. The design plan envisions a demonstrator which has the potential to act as an interface to the actual TDS. Through the use of hypermedia, the simulation environment will allow users to view varying levels of detail according to their own interests. The result will be a simulation environment which can be either exported or used locally in a highly interactive format. Software and hardware considerations and recommendations are also included.

TDS provides cost and capacity information on specified job and training patterns by simulating the flow of airmen through them.

14. SUBJECT TERMS
computerized simulation
cost estimates
demonstrations
military training
training decision making

15. NUMBER OF PAGES
28

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

TABLE OF CONTENTS

TRAINING DECISIONS MODELING

SIMULATION ENVIRONMENT:

DESIGN PLAN & HARDWARE OPTIONS AND RECOMMENDATIONS

1.0 INTRODUCTION

Project Background

Air Force training planners must determine the most effective and efficient ways to train people in the various Air Force Specialties (AFSs) using formal training courses, on-the-job training, mobile training, contractor courses, and various other training options. A Training Decisions Modeling methodology is being developed by the Air Force Human Resources Laboratory to provide a more integrated and unified approach for making training planning decisions. The multi-year research and development effort led by McDonnell Douglas to design and build the Training Decisions System (TDS) was the initial step in bringing together different information and trade-off requirements into an integrated decision support system.

Search Technology's current project in the Training Decisions Modeling program complements and builds upon the McDonnell Douglas TDS work. The project involves designing a highly interactive demonstration system that will embody more of the evolving capabilities of the Training Decisions Modeling Methodology. This effort will make the visions and benefits of the Training Decisions Methodology more apparent to potential users and program supporters.

Plan for this Report

This report reviews the requirements for the TDS demonstration system and the "exportable" version of it as presented in the original statement of work. It interprets these requirements in light of the work carried out during the familiarization and evaluation phase of this project.

Next, it presents the architectural and hardware/software implications of the requirements, identifying various options and making recommendations that best satisfy the requirements given the constraints of the installed base and tool capabilities.

Finally, the report outlines a design for the TDS demonstration system and a preliminary design plan.

2.0 REQUIREMENTS FOR THE TDS DEMONSTRATION SYSTEM

The initial requirements for the demonstration system were specified in the statement of work. The purpose of the first phase of the current Search Technology project was to become familiar enough with TDS to interpret and extend these proposed requirements. So that they can be easily referred to later in this document, the requirements are listed and numbered in the sections that follow (e.g., "R1" refers to the first requirement).

Requirements Specified in the Statement of Work

The primary requirement for the TDS demonstration system is that it accurately reflect the actual and potential capabilities of the TDS software being developed. The architecture of the TDS programs is commonly portrayed as four interrelated subsystems, the Task Characteristics Subsystem (TCS), the Field Utilization Subsystem (FUS), the Resource/Cost Subsystem (RCS), and the Integration and Optimization Subsystem (IOS). Hence, the statement of work lists requirements in terms of these four subsystems.

TCS Capabilities

The primary purpose of the Task Characteristics Subsystem is to identify Task Modules (TMs), which are collections of tasks that serve as a common representation for "what is done on the job" and "what is taught" that makes it possible to predict proficiency at the former from the latter.

The demonstration system must be capable of providing:

R1) TMs for requested AFSs

R2) Tasks for the requested TMs

R3) Most preferred training settings for the TMs

R4) Alternative training settings for TMs

R5) The training settings that yield maximum gains in proficiency

R6) Current training times for each training setting

R7) Optimal training times for each training setting

R8) Minimal training time (should compression of training occur).

FUS Capabilities

The Field Utilization Subsystem is the name given for the various data collection, analysis, and modeling activities involved in modeling the sequence of jobs and training courses that an airman goes through in a particular specialty or sub-specialty during an Air Force career. The representation of jobs, training, and the transitions among them is called a Utilization and Training Pattern or U&TP.

The demonstration system must be capable of providing:

R9) Current U&TPs for an AFS

R10) Sample alternative U&TPs for an AFS

R11) Management preferences for U&TPs

R12) Training requirements and proficiency state requirements for jobs in terms of the TMs comprising those jobs.

RCS Capabilities

The Resource/Cost Subsystem is concerned with the availability, capacities, and costs associated with the delivery of training.

The demonstration system must be capable of providing:

R13) The types of resources required for training a TM

R14) Estimates of the quantity of each type of requisite resource needed for training

R15) A compiled listing of the estimated resource types and quantities

R16) Estimates of the types and quantities of resources available at each training setting

R17) Estimates of an individual site's capacity to provide training on different combinations of TMs to specified numbers of personnel

R18) Estimates of the cost of providing training on each TM in each setting

R19) Estimates of the variable costs of providing training to different numbers of personnel on different combinations of TMs in different settings.

IOS Capabilities

The Integration and Optimization Subsystem (IOS) is the part of the TDS that relates the TCS, FUS, and RCS subsystems when simulations are carried out. The "user interface" to TDS, in the narrow sense of system inputs and outputs, resides in the IOS.

The demonstration system must be capable of simulating the analysis of at least 10 problems. The specific example problems to be used will depend in part on the availability of the relevant data, but typical problems might be like the ten that follow.

Example 1. What is the effect on proficiency of *increasing* by 10% the amount of classroom training on a particular TM?

Example 2. What is the effect on proficiency of *decreasing* by 10% the amount of classroom training on a particular TM?

Example 3. What are the cost savings of eliminating a TM from a particular AFS?

Example 4. What are the proficiency consequences of eliminating an existing training course?

Example 5. What are the consequences of consolidating several existing courses into a single more comprehensive course?

Example 6. What happens if the number of airmen entering an AFS is decreased by one-third?

Example 7. What is the impact of increasing the average tour length for certain jobs in a specialty?

Example 8. What is the effect of better testing policies that more systematically ensure that airmen who need a course take it?

Example 9. What is the effect of a 10% increase in re-enlistment rates on second-term proficiency and training costs?

Example 10. What are the consequences of moving training from a field detachment course or OJT to the basic resident course?

These simulated problems and their associated analyses will include:

R20) "Canned" answers for "what if" questions

R21) Simulated optimizations of U&TPs for cost, training time, and job performance

R22) Reports at various levels of detail that can be specified by the user to include information on questions being asked, the particular AFS under review, current and alternative U&TPs, and associated costs.

Additional Characteristics

In addition to these specific requirements to demonstrate specific capabilities of the TDS software and databases, the statement of work imposes six additional requirements. Five of these are related to the user interface of the demonstration system, which should:

R23) Be interactive and "user-friendly"

R24) Make full use of high resolution graphics

R25) Provide help screens

R26) Provide varying levels of detail on request

R27) Incorporate human factors design characteristics.

The final requirement identified in the statement of work is that the demonstration system should be designed to be extended in the future to reflect the planned integration of TDS with other training planning and modeling systems. Specifically, the TDS demonstrator should:

R28) Provide "hooks" for integration with other systems or simulations.

Requirements for an "Exportable" Demonstration System

In addition to these twenty-eight requirements specified in the statement of work for the TDS demonstration system, several other requirements are stated that pertain to an "exportable version" of the demonstration system. The exportable version must:

R29) Provide an executive overview of the complete demonstration system

R30) Demonstrate the general capabilities of the TDS at differing levels of detail

R31) Be fully executable on one Z-248, IBM PC/AT or compatible.

Interpretation of the Requirements

The purpose of the familiarization and evaluation phase of this project was to become familiar enough with TDS to be able to interpret and extend the requirements as proposed in the statement of work. The basic conclusion of that phase was that the TDS software is too complex to be understood apart from the context of the modeling methodology for training planning and analysis in which running the computer programs is a relatively minor step.

A second conclusion from the familiarization and evaluation phase of this project is that the intended audience for the demonstration system is diverse. The audience includes people who need only an executive overview of the training decisions modeling program, those who need to understand the modeling methodology, those who need to understand the capabilities of the TDS software, and those who need to be able to run the software.

At one extreme, the executive overview group of users needs only to understand the basic concepts of the TDS modeling methodology and has little need to understand what the TDS software does. At the other extreme, potential users of the TDS software need to learn about the specific inputs and outputs associated with the programs. Over time, the TDS software will become capable of supporting more of the TDS methodology, but it is essential that the relationship between the TDS modeling methodology and the TDS software under development always be clear to prevent users from confusing the complete methodology with the software.

The requirements for an exportable version of the TDS demonstration system (R29-R31 above) are intended to provide users who need only an executive overview of TDS to obtain what they need. However, this begs the question of whether it is possible to identify precisely what a user needs to know before providing either the complete demonstrator or the simplified overview version. It also assumes that someone who initially views the overview version will not later need to know that additional information contained in the comprehensive demonstration system.

Taken together, these observations from the familiarization and evaluation phase suggest that the initial requirements (R1-R31) might be enhanced as follows:

R32) The TDS demonstration system should explain what the demonstration system is supposed to demonstrate by presenting the TDS software capabilities in the context of th TDS modeling methodology. Put more tersely, the demonstration system should explain the definitions and concepts contained in R1-R22.

R33) The demonstration system should introduce the TDS modeling methodology in a way that does not require any exposure to or familiarity with the design or capabilities of the TDS software. In other words, the TDS executive overview should explain the modeling process without relying on architectural views of the software like TCS, FUS, RCS, or IOS.

R34) Incremental development of the TDS software by McDonnell Douglas implies that the TDS demonstration system must also be incrementally modifiable so that it can maintain an accurate distinction between TDS modeling concepts and TDS software capabilities.

Making the Demonstration System the TDS User Interface

However, if the demonstration system can explain the TDS software capabilities in the context of the TDS modeling methodology (R32), then the demonstrator will be making the TDS software significantly easier to use. This fact, in conjunction with the requirement (R28) that the demonstration system should be designed to be extended in the future to reflect the planned integration of TDS with other training planning and modeling systems, strongly suggests that the demonstration system could be designed so that it could later be extended to serve as the user interface to the TDS software. This would mean that experience with the demonstration system would completely transfer toward using the TDS software, and the demonstration system could more transparently track changes in the capabilities of the real system.

Making the Exportable Version Unnecessary

Note also that if the last two requirements (R32 and R33) for the demonstration system can be met while making it fully executable on a Z-248, IBM PC/AT or compatible, then the need for a separate exportable version of the demonstrator (R29-R31) vanishes. A complete demonstration system that can provide any level of detail required is far more useful than two separate systems. In addition, a single system with a consistent interface to both overview and detailed information is preferable to two systems that have a different "look and feel." Finally, if one comprehensive and adaptable system can be meet the requirements of two systems, the effort that would otherwise go into testing, packaging, and documenting the secondary system can be directed toward making the single system better.

Conclusion: A Single Demonstrator Design for Three Purposes is an Attractive Goal

It appears, therefore, that it is worth trying to design the TDS demonstration system to potentially serve as the user interface to the TDS software while making it portable and flexible enough to eliminate the need for a "stripped-down" exportable version. However, this goal may be too ambitious given the resources allocated to the demonstration system project, and it is important to note here that satisfying the demonstrator requirement is more important than building a prototype user interface to the TDS software.

3.0 ARCHITECTURAL AND HARDWARE/SOFTWARE IMPLICATIONS

Achieving this ambitious long term goal will require a basic architectural change in the TDS software and raises some challenging hardware and software selection issues.

Separate the TDS "Front End" and "Back End"

The first architectural change will be to separate the "front end" of the programs that contain user interface functions from the "back end" of the programs that execute the simulation. Separating the user interface from the simulation allows the demonstration system to present the user with the logical structure of input and output information, which is unlikely to change, and to ignore the complex data formats and multiple files required by the simulation programs, which are likely to change as the system continues to evolve.

Database as "Integrator/translator" Between Front and Back Ends

The optimal way to connect the front and back ends is through a database that is the repository of input and output data in a neutral format. Information about jobs, courses, transition probabilities, and other TDS entities is currently contained in a number of input files, which makes it tedious and error-prone to change any of it to run a new model. Each new model requires a new set of data input files, even though much of the information may not have changed. Storing information about courses, jobs, and other TDS entities in a database would minimize redundancy and increase maintainability and reuse. For example, many of the data items in the input files are probabilities that must sum to 1.0; changing any of them could automatically rescale the others to preserve this relationship.

The front end (which is either contained in or connected to the demonstration system) can format these data appropriately for presentation to users, while the back end formats these data appropriately for the simulation software.

McDonnell Douglas input is critical in establishing the specific capacity and performance requirements for this database if evolving the demonstration system into the TDS user interface turns out to be a realistic objective. Initially, the demonstrator will probably substitute flat files that contain "canned" input and output data for the database.

Hardware and Software Alternatives

These two desirable architectural changes imply that the current development and delivery environments for TDS may need to be changed. Several alternatives exist; while the McDonnell Douglas TDS software development project will ultimately determine which is selected, it is useful to consider the possibilities from the perspective of the TDS demonstrator and potential enhancements to the TDS user interface.

Current situation

The TDS simulation software is currently being developed by McDonnell Douglas in FORTRAN on a Digital Equipment VAX minicomputer and then ported to a Unisys mainframe computer at the AFHRL. Even now, this is a suboptimal solution, since the Unisys computer is neither readily accessible nor easy to operate for AFHRL. Moreover, the Unisys is probably incapable of supporting the user interface needed to make the TDS software easy to use.

Unisys as the back end

It might be possible to implement the front end functions on a PC or 386-based processor that communicates with the mainframe, leaving the back end simulation programs on the mainframe. This alternative still requires the porting step for the simulation software which could potentially be eliminated if the development and delivery platform were the same.

Workstation for both front and back ends

The primary requirement for the back end is that it have the capability and power to run the TDS simulation programs. Workstations like those provided by Sun, Digital Equipment, Data General, and other manufacturers were designed for scientific and engineering computing and are well suited for the TDS back end. Many databases are available (e.g.. Ingres, Informix, Oracle), and a variety of user interface tool kits could be used to implement a modern user interface for the front end.

The biggest drawback to the use of a workstation for TDS is that they are not common among TDS users, potential users, or others who might be interested in learning about TDS.

Apple Macintosh for both front and back ends

The Apple Macintosh is a superb software and hardware environment for graphical user interfaces. It supports a wide variety of user interface development tools that are well suited for TDS and the TDS demonstration system, especially HyperCard and SuperCard. In addition, many databases run on the Macintosh, including some that are noted for integrated user interface development environments (e.g., 4th Dimension, Wingz).

The Mac, like all Apple computers, receives less attention for its scientific computing capabilities, but nonetheless provides several options for FORTRAN compilers. The Mac II certainly has the computational power to run the TDS software.

The Macintosh's primary drawback for TDS is that it is not readily available to potential users. It may be more common than workstations, but not significantly so.

386-based processor for front and back ends

A final alternative architecture is to use a 386-based processor to host both the front and back ends and the integrating database. Numerous databases appear to be suitable (e.g., Paradox, Foxbase, RBase, Dbase, Oracle), and several user interface prototyping toolkits are available (e.g., Clarion, Guide, Layout, LinkWay). While some of these latter programs are being touted as "HyperCard for the PC," none yet appears to have comparable power or ease of use. On the other hand, this is an extremely volatile market segment for application software, and new programs and enhancements to existing ones are rapidly emerging. Selecting the program that is most likely to emerge from the pack of DOS contenders is an urgent and critical task for the development phase of this project.

The 386 alternative has several advantages over the others. The most important one is that 386-based processors are either available or likely to be available to most of the potential users of TDS and the TDS demonstration system. A related benefit is the likelihood that software developed for a 386-based processor can also run on 286-based processors, which are already ubiquitous in the user community.

One uncertainty with this alternative is whether it is better to use DOS or OS/2 as the operating system. OS/2 might support better user interface technology, but the desirability of eliminating the separate "stripped-down" demonstrator for the Z-248 by making the fully-developed demonstrator inter-operable between the 386 and 286 points to DOS rather than OS/2.

4.0 DESIGN

Design Concepts

Several design concepts follow from an analysis of the requirements and the capabilities of the hardware and software environment in which the TDS demonstrator will most likely be developed.

Present multiple interrelated perspectives, and make it easy to move between them ("hypertext")

The demonstration system must present TDS concepts, TDS modeling methodology, TDS software capabilities, and sample data from AFSs. Each of these presentations is primarily appropriate for different purposes and users. Nevertheless, while each perspective can be considered separately, each is enhanced by relationships with the others.

Computer-support for following the non-linear connections or relationships between different pieces of information is the definition of hypertext. Hypertext can be thought of as the computer analogue to the footnotes and cross references in printed documents that readers can follow for more detailed or related information.

Hypertext design allows users to select from a wide variety of potential pathways through the information. This freedom is often more engaging, motivating, and entertaining than user interfaces that present only a fixed number of ways to view information.

Hypertext concepts encourage modularity and the elimination of redundancy in stored information because information can be stored only once but viewed in any appropriate context.

Reuse information (transparently) rather than repeat it

For example, regardless whether users were primarily interested in the TDS modeling methodology, TDS software, or AFS data, they might still need definitions of basic TDS concepts (e.g., TM, U&TP, allocation curve). Instead of duplicating these definitions wherever they were useful, they should be collected into a TDS concepts Glossary and stored only once. Likewise, a definition of "U&TP" in the concepts Glossary can be enhanced by an example that contains actual jobs and training events, but this example should be reused from the AFS database rather than statically frozen in the concepts Glossary.

The primary function of the database in the proposed TDS architecture is to support reuse of input and output information. The same information can be viewed in different contexts by "projecting" it from the database at part of various TDS entities. For example, the probability of an airman's assignment to a training course can be viewed as part of a complete U&TP, associated with the course, or associated with the job.

Note, however, that reuse of information is only viable if it does not penalize the user. If it is tedious for a user to find the single copy of information, it is better to repeat it wherever it is needed. Hence, the design concept here is for transparent reuse of information made possible by rapid retrieval and display of the information in whatever context it is needed.

Support the progressive display of detail

"Progressive display of detail" is a design concept that is a special form of hypertext that will enable the demonstration system to meet the diverse needs of different users and make unnecessary a separate "stripped-down" exportable version. Users can be presented with a high-level or outline view of the information contained in the demonstration system. But unlike an outline on paper, this outline view on the computer can be dynamic so that the user can select headings or topics and progressively expand and contract details to view only the information that is of interest.

This approach can be thought of as a generalization of the idea of separate "tracks" for expert or novice users of the system. Instead of a single implicit or explicit option to view either overview or detailed information, or even more coarsely, to use either the complete demonstration system or a stripped-down version, users can continuously decide on the level of detail they desire at any point.

A user interface that embodies this concept of progressive display of detail makes "online help" an intrinsic component. Each TDS entity, report, or data object can have associated with it additional information or help that can be viewed in context.

Design Details

Design Metaphor: "Interactive book"

Many aspects of the design concepts in the previous section point toward the metaphor of an "interactive book" for the user interface of the TDS demonstration system. That is, the user of the TDS demonstration system will learn about various aspects of TDS via reading or browsing an "interactive book." In some respects, a table of contents for a book is nothing more than a list of topics. However, the book metaphor is more familiar, engaging, and graphically oriented than a simple list, and suggests a more consistent set of characteristics and expectations for the user interface than a list does.

The table of contents for the TDS demonstrator "book" lists four main topics:

* TDS concepts (Glossary)
* TDS modeling methodology
* TDS software
* AFS databases.

Like book chapters, each of these topics is primarily hierarchical and contains sections and subsections. Also as in printed books, each TDS "chapter" is enhanced by numerous cross references to other chapters.

Printed documents and books often contain a variety of "structure landmarks" that locate important parts. Tabbed dividers, color coded pages, or characteristic layouts for different sections all help readers know where they are. Analogous screen conventions for each of the four TDS "chapters" can serve the same purpose.

Printed documents use various typographic conventions to signal relationships and cross references of different types. Bold type is often used for the first appearance of a Glossary term. Underlining or italics customarily indicate the names of cited books or products. It should be possible to use conventions like these to identify for the users terms that have associated Glossary definitions, example data, or other related information.

All of these design ideas so far can probably be implemented given the available resources for the project. Properly designed, however, the demonstration system can continue to be enhanced in the future to employ additional aspects of the book metaphor to further increase its usability. One possibility is to enable individual readers to customize their version of the demonstration system, just as readers customize their printed books with margin notes and bookmarks.

Readers often create their own navigation aids in the course of using books. These include explicit location markers like bookmarks and turned-down pages, as well as implicit location markers like dirty and coffee-stained pages that signal "you've been here before."

The TDS "interactive book" can contain analogues to these navigation features that will help users know where to go and to return to where they've been. In a computerized book, these are easy to implement as a stack of previously displayed pages for "backtracking" for in-order return. Changing this stack of previously-viewed units into a menu turns a backtracking facility into a history or bookmark list that supports out-of-order return to any previously-viewed page.

Browsing the "TDS book" -- Examples

Example 1. Suppose the user of the TDS demonstration system begins by selecting the topic of "TDS modeling methodology" from the table of contents on the opening screen. The screen changes to display the first "page" of the TDS modeling methodology chapter. This page contains some introductory text about modeling as well as additional headings for "sections" of the methodology chapter:

* modeling AFS career paths
* modeling tasks and training
* modeling costs and resources
* "what if" & optimization analyses.

Readers could also learn about the TDS modeling methodology by "zooming" into a graphic flow chart of the 26-step modeling methodology that appears in the TDS user manual. Like the "exploded diagrams" that often appear in engineering drawings, this graphic perspective will help users understand how the various steps fit together by viewing them at different levels of abstraction.

After selecting (from either the table of contents or the flow chart) one of the high-level concepts in TDS modeling like "modeling AFS career paths," the user begins reading about the steps involved in modeling AFS career paths. The discussion introduces the term **Utilization and Training Pattern** which, as it is here, is highlighted in bold type to indicate that a precise Glossary definition can be displayed by selecting the term.

The first time that readers of the TDS demonstration book encounter a highlighted phrase, they are likely to select it. This causes the definition to appear in a "pop-up" window that overlays but does not hide the background page.

After studying the definition, the reader may decide to view the actual U&TP from one of the AFS databases contained in the demonstration system. This is accomplished by selecting the italicized cross reference (*See example U&TP for AFS 328X4*) at the bottom of the screen page.

Example 2. Another user of the TDS demonstration system may view some of the same information but from a different perspective. A person who is studying to use the TDS software may initially select the topic of "TDS software" from the table of contents on the opening screen. The first page of the TDS software chapter presents the familiar architectural diagram of the TDS software with labels for the TCS, FUS, RCS, and IOS subsystems. If the user is familiar with these terms, there is no need to "pop-up" the definitions from the Glossary, and let's suppose instead that the user selects the FUS topic for more detail about the software design.

The description of FUS subsystem mentions the input files that are required by the current version of the software. The reader is familiar with the data formats for the files, but does not know where the data are supposed to come from. To learn about the steps in the TDS modeling methodology that are involved in creating the input data, the user follows the italicized cross reference (*See associated modeling steps*) at the bottom of the screen page.

Sample reports could be contained in an Appendix to this part of the TDS interactive book and browsed as a collection. Alternatively, the sample reports could be "retrieved" in context from other parts of the demonstration system.

Design Plan

The following is the current plan for creating and implementing the design for the TDS demonstration system.

Select front end software

This task will determine the feasibility of much of the proposed design. The initial goal is to locate a user interface development toolkit that runs in both the 286 and 386 processor environments. Ideally, this user interface package will support ready integration with a database. Even if the appropriate database capability cannot be found, it may still be possible to build a demonstrator system that meets all other requirements except easy extensibility as the user interface to the TDS software. Since the demonstration system takes priority, a more critical problem at this stage would be the failure to find appropriate user interface tools that easily support hypertext functionality.

Design and implement prototype front end

This task will maximize the reuse of effort and software, and obtain early and useful feedback on the appearance and functionality of the demonstration system. Rather than develop the proposed design as a set of static screens, this task develops an executable prototype that shows the complete "look and feel" for part of the system instead of just the "look."

Revise prototype and incrementally extend to create demonstrator

After the review of the prototype system, the complete demonstration system user interface can be implemented down to the "hooks" to the actual data.

Implement back end and load sample data

If an appropriate database can be found in the initial task of this design plan, this is a large step. Otherwise, this step involves data collection and organization as a set of example files attached to the front end. This latter alternative is tedious but less difficult than the former of designing a database schema and loading the data.

Integrate front and back ends

If a database is involved, this task involves writing the programs that "project" the TDS examples by retrieving data from the database. Otherwise, it involves creating sample reports by "hand-crafting" static examples.

Revise integrated system

The complete demonstration system can be revised after review by the program sponsors.

5.0 SUMMARY STATUS OF PHASE 1 WORK

This design plan is the final task in phase 1 of the TDS demonstration system project. It will be presented to the program manager on March 27, 1990. The final contract deliverables for phase 1 will be the presentation materials and conference minutes from that briefing.