

Contract No. F33615-88-C-3612
Report No. NAWCADWAR-92087-60



GUIDELINES FOR ADAPTIVE AID DESIGN: HYPERTEXT REFERENCE SYSTEM FUNCTIONAL SPECIFICATION

Robert C. Andes, Jr.
SEARCH TECHNOLOGY, INC.
4725 Peachtree Corners Circle, Suite 200
Norcross, GA 30092-2553

25 NOVEMBER 1991

Interim Report
Period Covering 22 July 1991 to 25 November 1991

Approved for Public Release: Distribution is Unlimited.

19960419 118

Prepared for
Air Vehicle and Crew Systems Technology Department (Code 6021)
NAVAL AIR WARFARE CENTER - AIRCRAFT DIVISION WARMINSTER
P.O. Box 5152
Warminster, PA 18974-0591

NOTICES

REPORT NUMBERING SYSTEM - The numbering of technical project reports issued by the Naval Air Warfare Center, Aircraft Division, Warminster is arranged for specific identification purposes. Each number consists of the Center acronym, the calendar year in which the number was assigned, the sequence number of the report within the specific calendar year, and the official 2-digit correspondence code of the Functional Department responsible for the report. For example: Report No. NAWCADWAR-95010-4.6 indicates the tenth Center report for the year 1995 and prepared by the Crew Systems Engineering Department. The numerical codes are as follows.

Code	Department
4.1	Systems Engineering Department
4.2	Cost Analysis Department
4.3	Air Vehicle Department
4.4	Propulsion and Power Department
4.5	Avionics Department
4.6	Crew Systems Engineering Department
4.10	Conc. Analy., Eval. and Plan (CAEP) Department

PRODUCT ENDORSEMENT - The discussion or instructions concerning commercial products herein do not constitute an endorsement by the Government nor do they convey or imply the license or right to use such products.

Reviewed By:  Date: 6/8/95
Author/COTR

Reviewed By:  Date: 6/8/95
LEVEL III Manager

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 25 November 1991	3. REPORT TYPE AND DATES COVERED Interim 7/22/91 - 11/25/91	
4. TITLE AND SUBTITLE GUIDELINES FOR ADAPTIVE AIDING DESIGN: HYPERTEXT REFERENCE SYSTEM FUNCTIONAL SPECIFICATION		5. FUNDING NUMBERS C: F33615-88-C-3612 Subcontract: 1146	
6. AUTHOR(S) Robert C. Andes, Jr.			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) SEARCH TECHNOLOGY, INC. 4725 Peachtree Corners Circle, Suite 200 Norcross, GA 30092-2553		8. PERFORMING ORGANIZATION REPORT NUMBER STI-TR-9103-002	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Vehicle and Crew Systems Technology Department (Code 6021) NAVAL AIR WARFARE CENTER-AIRCRAFT DIVISION WARMINSTER P.O. Box 5152 Warminster, PA 18974-0591		10. SPONSORING/MONITORING AGENCY REPORT NUMBER NAWCADWAR-92087-60	
11. SUPPLEMENTARY NOTES NAWCADWAR POC: LT Meagan Carmody (Code 6021)			
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for Public Release; Distribution is Unlimited.		12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) Preliminary specifications for a hypertext-based adaptive aiding design reference are given. The system will provide a repository for the design and empirical validation of adaptive aiding system designs. Particular emphasis is placed on highlighting design guidelines and empirical support for aiding recommendations. The specification aims to capitalize on design information and guidelines extracted under another AFAIC task. Data types, input/output, and information seeking behavior by designers consulting with the hypertext system are addressed. Particular emphasis is placed on extensibility of the system as more research is conducted in adaptive aiding. User profiles, and recommended hardware and software for system implementation are discussed. A preliminary implementation schedule is provided.			
14. SUBJECT TERMS adaptive automation, adaptive aiding, hypertext reference system for the design of adaptive aiding		15. NUMBER OF PAGES 25	16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL

TABLE OF CONTENTS

1.0	Introduction	1
1.1	Background	1
1.2	Purpose of this Document	1
1.3	System User Profiles	2
1.3.1	Primary User Profile	2
1.4	Implications of User Profile on System Requirements	4
2.0	Hardware and Software Requirements	6
2.1	Reviewed Environments	6
2.1.1	Macintosh Hypertext Software Tools	6
2.1.2	Personal Computer Hypertext Software Tools	7
2.2	Development Environment	8
2.2.1	Recommended Hardware	8
2.2.2	Recommended Software	9
3.0	System Data	10
3.1	Data Types and Files	10
3.1.1	Enhanced Bibliographic Reference	10
3.1.2	Design Guidelines	11
3.1.3	Lessons Learned and Material Summary	12
3.1.5	Notes List	13
4.0	Functional Description	14
4.1	Basic Concept of Operation	14
4.2	Top-Level Functions	15
4.3	User Interaction Example	15
5.0	Specification of Functions	17
5.1	Navigation and Information Retrieval	17
5.1.1	Browsing	17
5.2	User Annotation	19
5.2.1	Links	19
5.2.2	Notes	19
5.2.3	Bookmarks	19
5.3	Create, Save, Manage, and Integrate Files	20
5.3.1	Open System	20
5.3.2	Close System	20
5.3.3	Select/Mark Material for Printing	20
5.3.4	Import	20
5.3.5	Quit	20
5.4	Integrate/Edit Files	20
5.4.1	Integrate Material/Links	21
5.4.2	Construct Links	21
5.5	User Interface Management Functions	21
5.5.1	Window Management	21
5.5.2	Material Views	21
6.0	Program Plan	22
7.0	Summary	23
8.0	References	24
	Notes	25

HYPertext REFERENCE SYSTEM FUNCTIONAL SPECIFICATION

1.0 INTRODUCTION

1.1 Background

Interest in computerized aiding systems to assist the operator of complex systems (e.g., process control, tactical aircraft) has been increasing steadily over the last ten years (Rouse, 1988). This trend has been due to the increasing capabilities of control systems and complexity of operation. A type of automated task execution assistance, known as adaptive aiding, has been introduced as a way to help the operator execute tasks in these high workload environments.

Adaptive aiding technology -- flexible, computerized aiding systems that adapt to the human operator's needs according to the current operating demands -- is developing in response to increasing control and monitoring demands on the operator. Recently, the conceptual viability of adaptive aiding has been demonstrated (Rouse, 1988). However, given the lack of fielded systems and only a small number of research results, a standard approach to aid design has not been established.

Search Technology's current effort on the Adaptive Function Allocation in Intelligent Cockpits (AFAIC) program, of which this present effort is a part, is to catalogue the relevant literature from various disciplines (e.g., human performance, engineering psychology, systems design, human factors, etc.) and extract valid guidelines and lessons learned from adaptive aiding efforts. The current undertaking is an effort to transform adaptive aid system research into the realm of engineering design, rather than allow it to remain in a form only understood by basic researchers. In addition, a supporting goal of this effort is to identify where necessary basic and applied research ought to be conducted to fill in the knowledge required to produce useful, reliable adaptive aiding systems in various operating domains.

1.2 Purpose of this Document

This document describes functional specifications for a computer-based guideline reference system product intended to serve as a repository for information compiled during the AFAIC effort. The product is a hypertext database system containing literature references, design guidelines extracted from the references, lessons learned, and supporting empirical data relevant to aiding systems design. This product is designed to be a comprehensive reference system for aiding systems designers, in which all information affecting an aid design effort can be retrieved in a complete, logical, and easily navigated format.

Several core design documents have been reviewed to date. The complete bibliography of material to be contained in the first prototype of this system can be found in the literature review providing the initial information source for this system

(see: "Guidelines for Adaptive Aid Design : A Review of the Literature", Anoskey and Andes, 1991). Primary value added of the current effort is to arrange the guidelines, lessons learned, design implications, and empirical support into a format that would increase the likelihood that this information will be considered by the designer at the appropriate stage in the aid production process.

This document describes the Adaptive Aiding Design Reference System in functional terms. Implementation details shall be disclosed in a detailed design document.

1.3 System User Profiles

There are several levels of user possible for this system. For the first prototype, however, we will only focus on one user group. This user profile is described in this section.

The primary market for this product is the engineering design and behavioral science communities collaborating in the design of adaptive aiding systems. This is a highly limited market; however, functionality can be expanded for other markets as design knowledge is gained and more aiding systems are implemented. The following user description focuses on members of this market segment only.

1.3.1 Primary User Profile

1.3.1.1 Educational Background

Professionals who we expect to be primary users play a significant role in both top-level aiding systems design and generation of empirical results influencing the top-level designs. As such, we expect that these professionals will have attained at least a Master of Science Degree in either behavioral science or an engineering discipline. Perhaps up to 50% of the users will also hold PhD degrees. Basic knowledge of computers is assumed, given the nature of the technology.

The possible range of users suggested in the previous paragraph suggests that users will be generally familiar with all areas relevant to adaptive aiding design (e.g., workload, human factors, human performance, intelligent systems), but not specifically in the area of adaptive aiding.

1.3.1.2 Knowledge and Skills

It is anticipated that users will only possess moderate familiarity with the breadth and focus of the material contained within the product. As mentioned earlier, it is anticipated that the typical user will most likely be a specialist in one of the subfields contributing to the knowledge necessary for successful implementation of aiding systems.

That fact notwithstanding, users of the tool will need to know which guidelines from other contributing fields -- in addition to guidelines unknown in *their* specialty field -- affect the design problem at hand. This knowledge must be characterized in terms that the user can understand. Without a priori knowledge of each possible user's background, the knowledge contained within the tool must assume a medium familiarity with engineering terminology and also be (somewhat) familiar with jargon

indigenous with human performance and perception. A glossary of terms should resolve any misunderstandings at this level.

System designers often consider themselves to be representatives of the end-user community. Through introspection, personal requirements, and empirical results, they make their design decisions. This perspective must be considered in the design of the tool. Design approaches to implementing specific aid functions can be presented as lists of alternatives, or rather "equivalent" approaches that serve as advice instead of prescriptive information. In essence, the knowledge would be available as the bridge from concept to implementation with several possible approaches to the design available. In this way, the designer can evaluate alternatives without needlessly constraining the design before it is completely formed. This will facilitate the creative process necessary for new designs. The approach given here would be particularly valuable to the highly educated designer described in previous sections.

Users will most likely possess medium to high computer literacy. The literacy will not be evenly distributed over types of computing or operating systems. The majority of users will be familiar with PC environments for general user tasks and applications. Having this experience, the users will be more familiar with concepts and operations of the PC genre of computing environments.

The projected user population will probably want to import information pertaining to known guidelines or missing information relevant to aiding (e.g., new empirical results, expansion of established concepts). In order to do this, the computing platform and operating environment must be familiar to them. Most of the users will probably not be regular Macintosh users. However, the growing number of PC based products with graphical interfaces and Mac-like features may dissolve the large interface and operations differences between the two platforms over a short period of time.

1.3.1.2 User requirements

Projected users of this product will seek information at two levels:

- o Information and guidelines directly applicable to the design problem at hand, and
- o Information, guidelines, and empirical support for particular concepts relevant to design of state-of-the-art aiding systems.

Although at first these user requirements appear completely disparate, closer examination shows that the primary distinction is one of information depth. The former requirement is concerned with accepted rules and practices to be used in a current design; the latter is concerned with the verification and validation of these practices. The latter is also concerned with furthering the state of the art. Both information seeking behaviors should be considered in the tool since the user will probably engage in both types of behavior given a different purpose for access to the tool. Also, a user may require greater depth of information about a guideline if he questions its validity in design.

1.3.1.3 Summary User Profile

The previous discussion about user population yields the following summary profile:

Educational Background

Level of education Master of Science or equivalent
Professional specialty Behavioral science
. Engineering (system design)

Work Environment

Organization Government or aerospace engineering
Role Member of technical staff
Social environment Team member
Resource environment Moderate to extensive computing resources

Knowledge and Skills

Area specialty One of: human/machine interface designer, human factors, human performance, systems engineering
Computer literacy:
Hardware/software - general Medium to very high
PC - general Medium to high
Macintosh Low to medium
Graphical interfaces Medium to high
Hypertext interfaces Low to medium

1.4 Implications of User Profile on System Requirements

Based on the Summary User Profile, the following system requirements emerge:

1. Information and guidelines about human behavior in complex control tasks should be highlighted at engineering (rather than research) level.
2. All guidelines should have supporting empirical evidence and related information readily available with easy access (where possible). This is necessary to ensure that substantiated guidelines supplant user's personal opinions and experience when valid empirically based guidelines exist.
3. A glossary or dictionary should be provided to reduce confusion due to "jargonization".

4. The guidelines and information contained within the tool is augmentable with user's personal, domain, and problem specific knowledge.
5. The system should be easily expandable with new research results.
6. The tool should support export of information for design reviews.
7. Relevant Mil-Standards should be included where applicable. This tool will be used by government employees and aerospace designers who must observed applicable standards for design.

2.0 HARDWARE AND SOFTWARE REQUIREMENTS

2.1 Reviewed Environments

The hypertext reference system should be developed and delivered on a personal (e.g., PC-class or Macintosh) computing system. This approach will allow a large number of users to employ the system without requiring mainframe access. In addition, learning a new system (e.g., hypertext-based reference) will not involve learning a new operating system as well.

Two such platforms were reviewed: the DOS-based personal computer, and Macintosh running under the MultiFinder operating system. Hypertext development products meeting the criteria set below were reviewed for each platform. This is not an exhaustive review: The focus of this section is on low-cost hypertext development tools, rather than high-end hardware platforms and custom applications.

The top two hypertext development tools on each platform are described below. These choices are based on Glushko's (1990) article entitled "Using Off-the-shelf Software to Create a Hypertext Electronic Encyclopedia". Other information was ascertained from programmers possessing programming experience with hypertext development tools. Requirements for the software product included: price, searching and browsing features, ease of use, link programming, user interface, and programming features of the product (some of the information is taken directly from Glushko, 1990). Basic hardware configuration for the recommended platform is given in the next section.

2.1.1 Macintosh Hypertext Software Tools

The most popular hypertext tools for the Macintosh environment are HyperCard¹ (Apple Computer) and SuperCard² (Silicon Beach). They both exploit the desktop metaphor interface of the Macintosh, and as such, appear to be indigenous software systems on the Mac.

In part, this is true. HyperCard was originally packaged with the Macintosh; it was further enhanced with the release of HyperCard 2.0. SuperCard, on the other hand, was produced by Silicon Beach Software to overcome some of the limitations of HyperCard. It is more of a software development environment, complete with multiple window support, programming project maintenance facilities and better importing facilities than HyperCard. Both of these products are low-cost hypertext tools, but are only available for Macs. They are reviewed below.

Hypercard 2.0 - HyperCard is an object-oriented programming product that allows a user to easily translate an idea into a working computer prototype. It was the first widely used hypertext product. HyperCard uses a "card stack" metaphor resembling an electronic version of a stack of index cards. Users can construct "stacks" for virtually any subject from address indexes to complex hypertext implementations. HyperCard uses (in reverse aggregation order) stacks of cards, cards, buttons, fields, and icons to integrate into buttons. What makes HyperCard so attractive is that the user can do as little or as much programming as desired. Several example stacks, buttons, fields, etc. are available for cut and paste. The hypertext capabilities of HyperCard consist primarily of the ability to link cards to cards, fields to fields, and also other combinations of HyperCard objects.

Built-in visual effects like dissolves, zooms, and wipes enable a kind of animation when cards are displayed in rapid sequence. HyperCard is supported by an object-oriented programming language called HyperTalk. It can be extended by any utility that runs on a Macintosh.

In HyperCard, there are no limitations on the number or size of units. With HyperCard 2.0, multiple cards can be displayed at once (in contrast with HyperCard 1.2) but links cannot be made between all object types. For example, scrolling text cannot contain "hot" (clickable) terms, nor can scrolling text be linked to other cards. This limits the size of text per unit.

HyperCard can also integrate text and non-textual components. Bit-mapped graphics can be placed onto cards easily. Any part of a card can be the source of a link. Selecting a link can also enable sound, animation, etc. as well, giving HyperCard multimedia capabilities. It supports a limited (and slow) search facility, however this can be overcome with external products. It provides built-in navigation via "previous" and "next" buttons as well as index and a graphical representation of the most "recent" cards visited. Bookmarks, notes, help etc. can all be supported.

The major limitation to HyperCard is its ability to handle large applications. It is a complete programming environment with multi-media support, but requires an experienced programmer to customize it for a specific application. As such, "it is a superb prototyping tool for user interfaces and much less suited as a delivery platform for large hypertexts" (Glushko, 1990).

SuperCard - SuperCard, on the other hand, does not use the card metaphor, and anything can be a button. SuperCard "projects" (series of files, etc.) contain windows (scrolling windows, title box, dialog box, etc.) and windows contain cards. With SuperCard, the user is not limited to one stack of cards. A number of windows can be contained in each project. SuperCard has more extensive import and export facilities, in addition to better text, window, and graphics editing. The extensions to the HyperTalk language include a number of ways to reference an object (previously a problem), better handling of resources and no window formatting limitations. The most attractive features of SuperCard are that independent Macintosh applications can be built (not on top of HyperCard, etc.), compiled, and run. Editors enable the programmer to debug animations, import and enhance graphics, etc. SuperCard should be run on at least a Macintosh II (for speed).

2.1.2 Personal Computer Hypertext Software Tools

Guide - Guide³ (OWL International) is available as both a PC/AT version and a Macintosh version. Only the former is considered in this description. Guide is a Windows⁴ (Microsoft) application, so it can exploit the interface and mouse support of Windows.

Guide looks like Windows due to its interface requirements. The desktop metaphor of Windows applies completely to Guide. Articles (or separate text theme) are separate files (limited only by DOS file size) and can contain bit-mapped graphics. Graphics can be combined and be "exploded" to provide more detail. However, this capability is of limited value to the current system.

Guide's strong suit is a rich set of link types. Link types are represented by text attributes, cursor shape, or by explicit markers (Glushko, 1990). Hierarchical disclosure, or replacement of text with explanation, for example, is supported.

Reference links are supported for direct access, as are note links for detail. Command links activate a command hidden in a definitions window, while expansion links allow the display of hidden "detail" information about a higher level topic. Two versions of Guide are present: one for authoring, one for just navigating. It is a simple matter to switch from one to the other.

Guide supports tables of contents, and hierarchical organization of documents and ideas. Limited string search and full-text search are available but are rather slow. Guide maintains a stack of areas visited and provide navigation assistance to the user. Navigation within Guide is limited however, bookmarks on documents are not supported. Nonetheless, Guide has some of the best features found in a PC product.

Two programming languages are available in Guide: Logiix and OPCL3. OPCL3 is designed to handle general Guide document handling (e.g., automatic document close, auto reformat, etc.). Logiix, on the other hand, is a more powerful language. It allows the programmer to activate data exchange between applications, write complex scripts (like HyperTalk), run dialogue, etc.

One particular capability of the current release of Guide makes it attractive: automatic import and basic indexing of dBase files (i.e., the current Adaptive Aiding Guidelines Database is in *.DBF format). This feature allows the implementor to directly import the work that has been completed thus far. Though a great deal of work must be conducted to complete the product, Guide's import facility allows the team to build on complete work rather than beginning over. A major drawback to Guide is document access speed. Additionally, a powerful hardware platform is necessary for user acceptance.

ToolBook - Toolbook⁵ (Asymetrix Corp.) is basically a PC-based version of Hyper/SuperCard (several of the HyperCard features discussed earlier also apply to ToolBook). It too runs under Windows 3.0. ToolBook uses a book metaphor as its basis. The user opens a "book", and each separate segment of data or graphics is contained on a "page". Tables of contents are easily done, as are simple navigational aids, very much like HyperCard. The scripting language available in ToolBook resembles both HyperTalk and Visual Basic.

Toolbook's strong suits are its rapid prototyping (like HyperCard), graphics import, and scripting capabilities. It also imports dBase files, but with limited functionality. Complex cursor control and access speed are drawbacks.

2.2 Development Environment

The following development and delivery platform is recommended:

2.2.1 Recommended Hardware

- o Personal Computer 386 PC (or 100% compatible)
 - minimum 25 MHz clock speed
- o 1MB RAM or higher
- o VGA graphics adapter and card

- o Standard keyboard
- o Microsoft (or equivalent) mouse
- o 1 - 720 Kb Floppy Disk Drive
- o 1 - 40 MB Hard Disk Drive (2.5 MB for hypertext tool)

2.2.2 Recommended Software

- o DOS 5.0
- o Guide (version 3.01) \$495. Requires Windows 3.0.
- o Windows 3.0 (\$90)

This configuration is specified for the following reasons:

1. The PC platform is most commonly used and understood personal system in use.
2. The Windows desktop metaphor is familiar to most computer users.
3. Guide is a highly flexible development environment *specifically* designed for hypertext applications. As such, it supports product development and has better programming features. Environment support provides many of the desired system functions without further programming.
4. Guide is portable to many other hardware and software platforms (e.g., Unix operating system, mini-computers, etc.). It can interface directly with many other software tools as well (e.g., dBase, Oracle, Sybase, etc.). We may desire to provide direct input to the system from on-line databases in the future.

3.0 SYSTEM DATA

3.1 Data Types and Files

Five basic data types will be contained in the Adaptive Aiding Hypertext reference system. Three types are primarily design reference information -- Enhanced Bibliographic References, Extracted Design Guidelines, and Material Summaries written from a design expert's perspective. This information will initially be directly imported from the database source files containing the information (this process has already been completed).

The last two types of information are system databases containing navigation and operational information. In the links database, all of the object links, by link type, present within the system are catalogued for fast reference and also used for system link verification when new material is added. The notes database catalogues all of the annotations present in the system and also contains run-time loaded, user added annotations to assist in interface customization. Term definitions are considered notes and are not a separate data type. All data types are explained in the following sections.

3.1.1 Enhanced Bibliographic Reference

The main objects within the system are the bibliographic references for each article included. Standard database record information is contained within each reference (e.g., full title, source, authors, abstract, etc.). In addition, the reference data contains information about the nature of the article (i.e., empirical investigation, concept development, etc.) and other insightful categorizations that will be helpful for the designer. The complete record information is given below. For each data item, the name, types, and other relevant information are listed. (This is based on database record format constructed during the literature review task).

The following information is associated with each record in the bibliographic database:

Data Name	Data Description	Data Types
ID	Unique identification symbol (system specific)	pointer
Title	Full title of article	character string (~ 255 characters)
Date published	Year	character string
Lit_type	Source literature type	character string: Book Journal article Conference proceeding Tech report
Domain	Application domain of material	character string: Aerospace or non-aerospace

Subject type	The primary theme of the material contained in the article	character string: Concept or architecture description Implementation Empirical investigation Literature review Evaluation of system or architecture
Biblio_ref	Full bibliographic reference source	character string
Pages	Number of pages	integer

3.1.2 Design Guidelines

The most useful information within the reference system are the design guidelines that have been extracted from reviewed articles. This information will be used in two ways: By system designers with design problems at hand and also for the researcher looking to identify areas where more research results are needed to provide direction for the system designer. As mentioned in the user profile section (1.3), these roles are often played by the same person.

Design guidelines are presented as just that: **guidance**. A useful guideline will discuss specific information, such as parameter values for models of human performance or particular contexts in which aiding should be employed. A concerted effort has been made to reduce the number of general, or nebulous, guidelines. This is because of the tendency of designers to be context insensitive in pressure design situations.

The guidelines information contains the guideline text itself, source of the information, and rationale behind the guideline. Further, they are categorized according to possible design issues to which they are most applicable (this is discussed in the literature review and indicated in the table below). The complete information description for guidelines is given in the following table.

Data Name	Data Description	Data Types
Ref. symbol	A pointer to the source article. The "relation" to the bibliographic database.	pointer
Author_ref	Reference to source (short)	character string
Date	Year	integer
Guideline_type	General categorization of guideline, e.g. Does it pertain to principles of interaction or principles of adaptation?	applicable to either: adaptation interaction

Contract No. F33615-88-C-3612
Report No. NAWCADWAR-92087-60

Applicable_aiding	Type of aiding most applicable for this guideline	character string: allocation transformation partitioning (func)tional allocation
Guideline	Actual guideline text	character field
Rationale	Support text for guideline	character field
Link1	Author reference (paper) for this guideline – conceptually related, empirical support for guideline, or direct reference (These are verification links for system implementation)	pointer
Link2	Second link slot, there may be more than 2	pointer
Framework question	Rouse (1988) proposed a framework for design of adaptive aiding systems. Each guideline is pre-classified according to the primary design issue that it addresses.	character string: What is adapted to? Who does the adapting? When does adaptation occur? What methods of adaptation apply? How is adaptation done? What is the nature of communication?
Implication	The guideline pertains to one of these four issues.	character string: performance workload user acceptance situation assessment

3.1.3 Lessons Learned and Material Summary

This information is the most subjective in the system. It attempts to summarize lessons learned (or lessons *to be* learned) as a result of reviewing this material. Additionally, the worth of the material to designers and researchers alike is assessed. The following information is contained in the lessons learned and summary data type.

Data Name	Data Description	Data Types
Ref. symbol	Pointer to source article	pointer
Lessons	Text of lessons learned in this article (if needed)	character field
Summary	Text summary of article's relevance to adaptive aiding written in an evaluative manner. This segment will also contain the tools & techniques information from the text representation. Evaluation as necessary.	character text

3.1.4 Link List

The links list file will be a database containing information about each link created for the system. Fields within this data type will include link name, link number, time, author, pointers (in a format to be determined) to each of the linked objects, and a pointer (TBD) to a file and text note that contains additional text and graphics pertaining to this link (if necessary).

3.1.5 Notes List

The notes list is a database that will contain information about each note created for the database. Fields within these records will include note name, note number, time, author, attachment status, pointer (format TBD) to attached object, and pointers to attached files (if any) comprising the note.

4.0 FUNCTIONAL DESCRIPTION

4.1 Basic Concept of Operation

Consistent with general Windows application software, the hypertext reference system will adopt the desktop metaphor most users will find familiar. In addition, the familiarity with the interface should facilitate learning of the system. The system will present basic information about a source of adaptive aiding information; depth of information sought, and guidelines accessed. Experiential information associated with a specific reference will be available through progressive disclosure. This process will be totally under user control.

At the most basic level, system functions will be integrated through the use of the Windows desktop. Users will access the system and different data (file) types through standard window operations within the hypertext development tool. The development tool environment also provides run-time interface features; these standard features now become part of the "system interface".

Once at the system interface, users will be presented with a standard bibliographic reference listing (default listing). Users may browse the reference information, read abstracts and summaries, expand the bibliographic information, or list guidelines for a reference. Standard reference links allow the user to peruse all information extracted from a reference, browse the list of guidelines, and return to the reference directly.

Other approaches to browsing the system information are possible also. For example, users may only wish to access information about a specific design question that they have in mind. A pre-defined classification scheme (see Anoskey and Andes, 1991; "Guidelines for Adaptive Aid Design: A Review of the Literature, pp. 8-12) is available within the system. Basic aiding design questions (e.g., what methods of aiding apply?, etc.) are used as classifiers for guidelines and lessons learned. A user can access the system and address one or more threads in the course of pursuing alternative approaches to the current design.

Another pre-engineered organization scheme will be available. Users seeking information about only human/aid interaction or only systems engineering of aiding systems will find the alternative organization helpful. This is particularly useful for different phases of design (i.e., software system engineering vs. user interface design).

Guide provides different link types as described in section 2.1.2. These capabilities will be exploited within the system. Link types, projected use of links, and navigation by link type is described in section 5.1.1.7 "Traversing Links". The user may have the guidelines presented in framework format (see previous paragraphs) and mark/save only that information pertinent to the specific design issue. Once this process is complete, the user may wish to compile a listing of all bibliographic references for the guidelines in the save set. Consistent with the Guide run-time interface, the system will support functions unique to electronic documents including rapid information retrieval through query and search, linking, merging other documents, etc.

4.2 Top-Level Functions

Based on the user analysis in this document and our understanding of the user population, the system will support five major functions:

1. Retrieve information from and navigate within the adaptive aiding reference and guidelines information base. This information base shall include the five types of data described in section 3.0 and also term definitions, where necessary.
2. Import information and expand the current information base within the guidelines reference system. This function will allow the user to add references pertinent to adaptive aiding design, add guidelines, lessons learned, and summary objects. Additionally, facilities will be implemented to allow the user to add links from guidelines to source, source to summary, etc. Special features will allow the linking of guidelines at a "conceptual" level. Initial import of the data will be Logiix scripted: Header information in the material to be added to the reference system will provide pre-format and initial layout (either *.DBF or ascii-text). Once the pre-format is complete, it will be up to the author to provide more robust linking.
3. Annotate the information base according to personal needs. Annotations shall consist of personal notes (section 3.1.5) about information and guidelines. Annotations will include attaching notes to three types of data (sections 3.1.1-3.1.3) and establishing links between information instances for future recall.
4. Access utilities or objects external to the reference system without losing current session instance.
5. Manage the interface during system operation.
6. Save and print specific information (e.g., references, guidelines, summaries, etc.) from within the system. The first prototype will support only basic save and print functions.

4.3 User Interaction Example

An example scenario involving the use of the tool to access design information is described in this section. In this scenario, the system user is primarily concerned with a specific design issue. The following narrative should provide a better image of user-system interaction. All data, guidelines, etc. are representative of the type of information contained within the system.

Setting

The user is a systems designer for advanced tactical aircraft. In the course of designing the pilot aiding system, he becomes concerned with providing aiding for the pilot engaged in complex situation assessment and communication tasks. He accesses the adaptive aiding reference system with the question: "What is the best method of aiding to use when assisting the pilot engaged in a complex situation assessment/communication task?"

5.0 SPECIFICATION OF FUNCTIONS

5.1 Navigation and Information Retrieval

5.1.1 Browsing

5.1.1.1 *Designate file*

The user will designate a file to browse (typically an alternate bibliography file) under the top-level file menu. There will be three primary file types to browse (they will also be symbolically linked): bibliography (section 3.1.1), guidelines (section 3.1.2), and summaries (section 3.1.3). Other file/data types will not be available choices at this level. The default organization of the active file will be represented by the outline displayed when the file is ready to be browsed. Complete file format will be discussed in the detailed design document.

5.1.1.2 *Expand/collapse (hierarchical only)*

Expansion link types within the Guide system support expansion and contraction of information (other tools support this functionality as well). This type of functionality will only be present when information about a particular topic/issue/approach, etc. is arranged hierarchically. Anticipated hierarchical arrangement will involve depth of information disclosure on a topic.

The user will activate this type of function to display additional headings or text under a main heading (e.g., design question to be addressed, etc.) in the browse mode. The heading is expanded/collapsed by clicking on the hot item. The shape of the cursor changes when expansion information is available for a topic.

5.1.1.3 *Scrolling*

Information windows and especially article summary windows will support mouse/key activated scrolling for information that fills more than one screen.

5.1.1.4 *Alternate view selection*

A "view" command will be available at the top level to facilitate access to alternate views of the displayed material (primarily guidelines). View will support three modes (where appropriate): Design framework view, default (TBD) view, and interaction/adaptation view. A view mode indicator will show the user what mode the display is currently in.

5.1.1.5 *Find item*

Text string or design issue searching will be available within the currently active file. The Find command searches a selected area of a document for the specified text. Find performs a local search only beginning at the current point in the file to the end.

The user may employ the Find command in an active hierarchical outline. The command will execute a depth-first search until the first occurrence of the text is located. When located, the cursor is positioned at the end of the found search string

and the display is scrolled to that point in the file. Only simple searching will be available in this version of the system.

5.1.1.6 *Branching*

Users will be provided with several ways to jump/branch from one context to another. This will be accomplished through the use of different link types (e.g., command links, reference links, etc.). Branch operations will be "two-way": Once the user has examined the hard link, a button will return him directly back from the origin of the branch.

5.1.1.7 *Traversing links*

A link is a logical or semantic association between two objects (reference) and may be defined by authors of the system, users, or maintainers of the system (e.g., to add new material). By clicking on a hot link "button" the user can activate the linked object in one action. Specific link types supported by both the development environment and those that are specifically programmed for this system will be discussed in the detailed design document. Basic link types provided by the development tools are described in section 2.1. Link types will be employed in the following ways (other applications TBD):

Reference Links – These links will be used to hard link the bibliographic reference to the extracted guidelines, lessons learned, and summaries. They will be two-way links and current position will be tracked via a link traversal stack. Backtracking and jumping will be supported. Reference links will also be used to connect related articles together (e.g., human performance results, etc.) and for connecting guidelines pertaining to a specific design issue for which the guideline applies. Information referring to a specific article source will display the reference (e.g., "Morrison, et al., 1990") as a hot button for direct access.

Expansion Links – Used to expand information about a reference (e.g. abstract, source type, direct references). More "in-depth" information about a guideline, for example, will be hidden by expansion links. The user can click on the term, etc. for more in-depth information. Expansion links will also be used for accessing guideline rationale and lists of related guidelines.

Note Links – Note links will be used to describe the classification of a guideline (e.g., pertains primarily to "How to aid.", etc.) and for expert comment on the material, rationale, and/or reasoning behind inclusion of this material in the information base. This type of link will provide the user with a great deal of background information. Term definitions will also be referenced by note links.

Command Links – These links will be used for activation of included models (if any), access to related graphics (results plots, architecture block diagrams, etc.). Complete functionality TBD.

Should the customer select another hypertext tool for system implementation, link types can still be preserved, albeit without system enforcement of link types.

5.1.1.8 *Next, previous, first, last*

These are specifically defined "go to" links between system objects. They will be available throughout the system.

5.2 User Annotation

A feature of this system will be user defined annotations (via "notes") in the information base. Annotations enhance the information base by providing the ability to customize the information in a problem- or user-specific manner. Two types of annotation will be available: links and notes. Links will be available for establishing relationships according to all link types available in the system. Notes will allow the user to attach additional information to any object in the system.

5.2.1 Links

5.2.1.1 *Create link and add information to links database*

Users will create links (this is similar to the author adding new material) by selecting "link" option from the main menu. The user will simply click on the origin material, access the destination material with the link command active, and click on the destination. When this operation is complete, a dialog box will appear with prompts for information about the link. Once the dialog box is completed, this information will be appended to the links database described in section 3.1.4.

Once a link has been created, it will function similarly to other links in the system. The user can activate the second object by clicking on the first.

5.2.1.2 *Delete link*

If the user decides to decouple two objects, the user indicates the link and selects delete from a top level menu. Undo will be supported for this function.

5.2.2 Notes

5.2.2.1 *Create and attach note to object*

The user will be able to construct and attach a note consisting of text, (possibly) graphics, and (possibly) links at any time during system operation. Dialogue boxes will walk the user through the process once "construct note" is selected from the function menu. Once the note is complete and attached, similar information will be written to the notes database. Notes will not be searchable (subject to hypertext tool capabilities).

5.2.2.2 *Delete note and identifiers*

The user will be able to delete notes and identifiers with one action. Undo will be supported for last operation.

5.2.3 Bookmarks

Ideally, bookmarks should be supported for this type of system. However, Guide does not support such functionality. Should the customer decide on a different development package, bookmark functionality should be specified. Some rudimentary functionality of this type may be possible through scripting.

5.3 Create, Save, Manage, and Integrate Files

It is assumed that the user is familiar with DOS operations and basic Windows functionality.

5.3.1 Open System

The open command displays a Windows dialog box with a listing of available files in the system. The user can also change drives or directories to access files of choice. Selecting the "open" button will activate the selected file.

5.3.2 Close System

The Close command will close all active windows associated with the current system and deactivate the run-time environment -- back to Windows. If user changes were made to the file, the system will prompt to save/rename the file before closing.

5.3.3 Select/Mark Material for Printing

The user can Print any or all of the records in the currently active file. If a user defined organization was implemented or if the file is presented in alternate view (section 5.1.1.4), the file will be printed in that format. Print options will be selected from a Windows-based Print Options dialog box. Further print functions based on user defined searches, etc. TBD.

5.3.4 Import

The Import command will be available primarily for use with dBase or formatted ascii-text files of article review information (e.g., bibliography, guidelines, summary). This function is indigenous to most hypertext construction tools. File import will be accomplished by selecting the Import command from the top level menu. Some data selection and formatting will be conducted via scripted reader functions (import scripts looking for embedded formatting information in the file) and through a dialog box. Once the information is read, it will be up to the author (user) to format the information and link it with existing information. Further information on incorporating new material can be found in section 5.4 "Edit files".

5.3.5 Quit

The Quit command exits the system, and returns the user to the Windows application screen. If the user has made changes to the file without saving, a dialog box will query the user to save/forget the current file state.

5.4 Integrate/Edit Files

A top-level requirement for this system is that it be updatable by users. This requirement implies that the user can review new material pertaining to adaptive aid design, import it into the system and link it to related information. Thus the new material would be completely integrated with existing material. The following functions are specified for accomplishing that task.

5.4.1 Integrate Material/Links

This function will be accomplished through the following sequence: The user imports a file through one of the previously specified methods (see section 5.3.4).

5.4.2 Construct Links

Once the material is imported into the system, the user can use the prespecified types of links (section 5.1.1.7) to integrate the new material into the system. For example, reference links will be established between the bibliographic reference and guidelines extracted from that reference. Further, a reference link will be established between the summary/lessons learned text and the bibliographic reference.

Expansion links will be employed for providing progressive disclosure of lessons learned information, guideline rationale, abstracts, etc. Note links can be employed to disclose definitions of operational terms, for explanation of guideline wording, etc. Should there be any embeddable models, etc., command links can be used to activate the models while viewing the system.

The previous paragraph dealt only with linking of an individual review. The more difficult task with incorporating new material is conceptual linking with existing material. This requires knowledge of the classification framework, adaptive aiding technology, and system architecture. Specific information pertaining to integration practices will be covered in the detailed design document.

5.5 User Interface Management Functions

The system will adhere to the Windows interface design standard to the extent possible.

5.5.1 Window Management

Windows are opened according to actions performed by the user. Clicking on an icon, button, or menu item are operations that result in opening or closing windows. A window can contain any type of information (e.g., explanation - expansion, annotation - note, etc.). Most windows can be moved, scrolled, sized, and closed. These functions will be accomplished through standard Windows operations. Windows may be tiled or overlapped.

5.5.2 Material Views

As stated in section 5.1.1.4, different views (organization) of the information may be accessed by the user. Depending on the view desired (e.g., design issue, interface vs. software) the information will be rearranged to reflect that organization. Alternate views will be selected by mouse-clicking the appropriate top-level menu item. When an alternative view is selected, the user remains at the same place as before the selection. An icon in the lower right corner of the active window indicates selected view.

6.0 PROGRAM PLAN

In this section, basic tasks and time estimates for implementation are given. Time estimates are based on months after contract (MAC). Staff estimates are in number of full-time equivalents (FTE).

<u>Task</u>	<u>Timeframe</u>	<u>Staff</u>
Revise System Requirements Document	1 MAC	1 FTE
Produce Detailed Design Document	2.5 MAC	1 FTE
Information Format Engineering	4.5 MAC	2 FTE
Review Other Relevant Material	7 MAC	1-2 FTE
Produce Prototype	10 MAC	2 FTE
Test Prototype	11 MAC	.25 FTE
Construct User's Manual	13 MAC	1 FTE

7.0 SUMMARY

A preliminary approach to a hypertext-based reference system for adaptive aiding design information has been specified. This system will provide a repository for information pertaining to the understanding, design, and empirical validation of aiding system designs. The system specification capitalizes on the already reviewed and summarized information developed under the AFAIC program. Data types, input/output, and browsing of the material were addressed with particular emphasis placed on the extensibility of the system. Given the current level of information collected under the literature review process (Anoskey and Andes, 1991) an evaluation prototype of the system can be constructed in a straightforward manner.

An extensive review of the projected user population was given. The user population will engage in different information seeking behaviors, specifically, they will seek specific guidance on a design question or look for missing information about how to approach a design problem. All of these behaviors should be supported within the current specification.

Alternative software and hardware platforms were reviewed based on several selection criteria (e.g., price, portability, software functionality, etc.). The recommended platform for the hypertext system is a PC-386 class machine using the Guide Hypertext software package running under Windows 3.0. Several reasons for this choice were given. Other alternatives are available. Details of system functions were specified with particular attention paid to concept of operations and system capabilities past initial prototype. Finally, a rough design and implementation schedule was given.

8.0 REFERENCES

- Anoskey, A.M. and Andes, R.C. (1991). Guidelines for adaptive aid design: A review of the literature. Search Technology Tech. Report (STI-TR-9103-001).
- Glushko, R.J. (1990). Using off-the-shelf software to create a hypertext electronic encyclopedia. In *Technical Communications of the ACM*, First Quarter.
- Rouse, W.B. (1988). Adaptive aiding for human/computer control. *Human Factors*, 30(4), 431-443.

Distribution List (Continued)

	No. of Copies
Air Force Office of Scientific Research Bolling AFB, DC 20332-6448	1
Naval Aerospace Medical Institute Naval Air Station, Pensacola, FL 32508	2
FAA Civil Aeromedical Institute Oklahoma City, OK 73124	1
NASA Ames Research Center Moffott Field, CA 94035	1
NASA Johnson Space Center Houston, TX 77058	1
Dr. Don Jarson Drexel University Philadelphia, PA 19104	1
Commandant of the Marines Headquarters Marine Corps Washington, DC 20380	1
Naval Weapons Center China Lake, CA 93555-6001	1
Naval Air Force U.S. Pacific Fleet NAS North Island San Diego, CA 92135	1
Commander, Naval Air Force. U.S. Atlantic Fleet Norfolk, VA 23511-5188	1
Center for Naval Analyses 4401 Fort Ave., P.O. Box 16268 Alexandria, VA 22302-0268	1
U.S. Air Forces In Europe (USAFE) Office of the Command Surgeon/SG APO, New York 09094-5001	1
Mr. Robert Montgomery 125 James Way ETC Corporation Southampton, PA 18966-3877	1
Dr Jean-Michel Clere Lamas/CEV F-91228 Bretigny-SUR-ORGE CEDEX FRANCE	1

Distribution List (Continued)

	No. of Copies
Dr. David Glaister, Royal Air Force Institute of Aviation Medicine Farnborough Hants GU14652 England	1
Mr. Michael Paul, Defense and Civil Institute of Environmental Medicine 1133 Sheppard Avenue P.O. Box 2000 Downsview, Ontario, Canada	1
Air University Library AUL/LSE, Maxwell AFB, AL 36112	1
Armstrong Laboratory Strughold Aeromedical Library/TSK-4 Brooks AFB, TX 78235	1
Commander, USAF Armstrong Aerospace Medical Research Lab Wright-Patterson AFB, OH 45433	1
Commander, Naval Training Systems Command 12350 Research Parkway Orlando, FL 32826	1
U.S. Air Force Tactical Air Command Office of the Command Surgeon (SGP) Langley AFB, VA 23665	1
Naval Air Warfare Center Warminster, PA 18974-0591 (1 Copy for Office of Chief Aeromedical Scientist) (2 Copies for Code 0471)	3
Commander, Naval Air Systems Command Washington, DC 20361 (3 Copies for AIR-320R) (2 Copies for AIR-931H) (1 Copy for AIR-531B) (2 Copies for AIR-5004)	8
1299th Physiological Training Flight Malcolm Grow USAF Medical Center Andrews AFB, Washington, DC 20331-5300	1
Dr. Fred Buick DCIEM 1133 Sheppard Ave W. North York, Ontario Canada M3M 3B9	1

Distribution List

No. of Copies

Defense Technical Information Center	2
Bldg. 5, Cameron Station, Alexandria, VA 22314	
Chief of Naval Operations	4
Dept of the Navy, Washington, DC 20350	
(1 Copy for NOP-506N)	
(1 Copy for NOP-591)	
(1 Copy for NOP-05H)	
(1 Copy for NOP-09E)	
Naval Aerospace Medicine Institute	2
Naval Air Station, Pensacola, FL 32508	
Office of Naval Research	1
Code 1433, Washington, DC 20375	
(1 Copy for Code 1433)	
Naval Postgraduate School	1
Monterey, CA 93940	
AL/CA-CF	1
Brooks AFB, TX 78235	
(1 Copy for Dr. Russell R. Burton)	
AL/CFTF	1
Brooks AFB, TX 78235	
(1 Copy for Dr. John W. Burns)	
Air National Guard Support Center	1
Mail Stop #18, Andrews AFB, DC 20331	
(1 Copy for ANGSC/SG)	
Advanced Physiologic Training Unit	1
833 Med Gp/SGT	
Holloman AFB, NM 88330	
(1 copy for 833 Med Gp/SGT)	
Federal Aviation Administration	1
CAMI Library AAC 64D1, P.O. Box 25082	
Oklahoma City, OK 73125	
(1 Copy for Civil Aeromedical Institute)	
Headquarters, U.S. Air Force	1
Bolling AFB, DC 20332	
(1 Copy for Office of the Surgeon General (SGP))	
Dr. Bill Albery	1
AL/CFBG Bldg 29	
2245 Third St.	
WPAFB OH 45433-7008	