NASA-TM-109708 19940023266

94N27769

SSP 30540 Revision A

NASA-TM-109708

Human Computer Interface Guide

Revision A

July 1993

National Aeronautics and
Space Administration

Space Station Freedom Program Office
Reston, Virginia

FREEDOM

REVISION NOTICE

FOR

HUMAN COMPUTER INTERFACE GUIDE

REVISION A – JULY 1993

Incorporating SSCBDs:	BB003118, eff. 05–14–93 BB003681, eff. 05–28–93

Robert P. Wilson /s/ for 7/7/93

Executive Secretary, Date
Level II Space Station Control Board

CHANGE INSTRUCTIONS

1. File the attached in a three–ring binder as indicated below:
2. Please remove SSP 30540, dated June 1991, and replace with SSP 30540, Revision A, dated July 1993.
3. Sign and date this page in the space provided below to show that the changes have been incorporated, and file immediately behind the Revision and History Page.

Name of person incorporating changes Date

IMPORTANT NOTE

The Configuration Management Office maintains an index of all SSCB baselined documentation. The SSFP Baseline Activity Index and Status Report (BAISR) is updated weekly and is available in the SSFPMail *SSCB Conference* of the Group Conferencing System or the PALS collection *SSPWORK*.

REVISION AND HISTORY PAGE

REV.	DESCRIPTION	PUB. DATE
	BASELINE ISSUE (REFERENCE SSCBD NO. BB000682 EFF. 7-12-91)	06-91
A	REVISION A (REFERENCE SSCBDs BB003118 EFF. 05-14-93 AND BB003681 EFF. 05-28-93)	07-93

U.S. Gov t

SPACE STATION FREEDOM PROGRAM OFFICE
HUMAN COMPUTER INTERFACE GUIDE

JULY 1993

PREFACE

The Human Computer Interface Guide, SSP 30540, is a reference document for the Information Systems within the Space Station Freedom Program (SSFP). The Human Computer Interface Guide (HCIG) provides guidelines for the design of computer software that affects human performance, specifically, the human–computer interface.

This document contains an introduction and subparagraphs on SSFP computer systems, users, and tasks; guidelines for interactions between users and the SSFP computer systems; human factors evaluation and testing of the user interface system; and example specifications.

The contents of this document are intended to be consistent with the tasks and products to be prepared by NASA Work Package Centers and SSFP participants as defined in SSP 30000, Space Station Program Definition and Requirements Document. The Human Computer Interface Guide shall be implemented on all new SSFP contractual and internal activities and shall be included in any existing contracts through contract changes. This document is under the control of the Space Station Control Board, and any changes or revisions will be approved by the Deputy Director.

Marc Bensimon /s/ for	7/12/91
Deputy Director, Space Station Freedom Program and Operations	Date

TABLE OF CONTENTS

PARAGRAPH PAGE

APPENDIXES

APPENDIX PAGE

TABLES

TABLE PAGE

FIGURES

1.0 INTRODUCTION

1.1 IDENTIFICATION

The Space Station Freedom Program (SSFP) Human–Computer Interface Guide (HCIG) document is a reference document for the Information Systems within the Space Station Freedom Program. The HCIG provides guidelines for the design of computer software that affects human performance, specifically, the human–computer interface. A related document, NASA–STD–3000, Volume IV, Man–Systems Integration Standards (MSIS), provides standards for the development of all man–machine interfaces. Information from NASA–STD–3000, Volume IV, referenced in paragraph 4.0 of the SSFP HCIG is formally cited.

1.2 SCOPE OF DOCUMENT

The scope of the document is limited to describing the characteristics of the computer systems with which the users will interact and the user–centered constraints on the design of that system. The document centers around guidelines for all software that affects human performance, especially the presentation of information and the real–time interactions with the system by which users input information. Guidelines are also included for some characteristics of input devices (e.g., keyboard, mouse) as they relate to software development for the Human–Computer Interface (HCI). The scope of the document does not include either a detailed description of the user interface software architecture or the code itself.

1.3 PURPOSE AND OBJECTIVES

The purposes of this document are to describe the elements of the HCI for the SSFP's computer systems and to provide guidelines to be used in developing the software for the HCI.

The objective of the design of the HCI is to increase the usability of the SSFP computer systems. A system manifests its usability through the speed and accuracy with which the users can perform tasks with it; novices' ability to learn to operate the system, and sporadic users' ability to relearn to operate it; and all users' preference for operating the system (ref. 60).

The design of an HCI must take into account the characteristics of the users, the tasks that the users will perform, the users' familiarity with the task, and the features of the system that will affect user performance. The critical characteristics of the users include their overall computer experience, the frequency with which they will interact with the system, and their role in relation to system operations (e.g., system management versus end user).

An understanding of the tasks that the users will perform allows the designer to shape the human–computer interface to perform the tasks with the greatest speed and accuracy.

Knowledge of the system features that may affect a human's interaction with the system, such as system response time and memory, can also benefit the design of the interface.

The development of the HCI is one part of an overall human factors design and evaluation process. Additional information about this overall process can be found in Military Handbook, Human Engineering Procedures Guide, DOD–HDBK–763 (ref. 4).

1.4 STATUS AND SCHEDULE

This document is based on USE 1000, Version 2.1, December 1988, which was developed by the Human–Computer Interaction Laboratory at the Johnson Space Center in support of the SSFP.

Figure 1–1, Overview Of the Development Of The Human Computer Interface Guide, is a schematic representation of the process used to develop this document. The initial phase of the development process consisted of acquiring knowledge about the SSFP standards and requirements (e.g., NASA–STD–3000 and SSP 30000, Program Definition and Requirements Documents); the Freedom Station computer systems, focusing on the users, their tasks, and the system features; and the scientific literature related to human–computer interactions, widely accepted existing guidelines for HCIs, and generally accepted practices. The second phase involved analyses of the data from the first phase. In phase 3, the guidelines were written based on these analyses. The development of the guidelines involved a formal review process within the User Support Environment Working Group. In phase 4, the guidelines were implemented as HCI prototypes, and these prototypes, as well as various alternatives, were evaluated.

1.5 ORGANIZATION

1.5.1 AUDIENCE(S)

A principal target audience for the HCIG is developers of user interface system software for the SSFP computer systems: the Software Support Environment, the Technical and Management Information System, and Space Station Information System. Accordingly, the document provides concrete guidelines that can be implemented in the development process, as well as specific examples of how to implement the guidelines. The intent of this document is to emphasize HCI guidelines needed during the developmental phases of the software life cycle. In addition, the document also gives software developers a set of evaluation tools to determine whether the particular implementation of the guidelines that they choose is productive. (See paragraph 5.0, Human Factors Evaluation and Testing of the User Interface Software.)

1.5.2 READING THE HUMAN COMPUTER INTERFACE GUIDE

This subparagraph describes the rationale for the organization of this document and provides the reader with tips about how to read the document. Because an HCI is a complex system, the information regarding an interface could be organized in many different ways, all of which would be proper depending on the needs of the user of the information. Consequently, the paragraph divisions and subparagraph headings could appear to be arbitrary, especially if the author's approach to human–computer interaction is not the same as that of the reader's.

The organization of the guidelines that describe the HCI (in paragraph 4.0, Guidelines for Interactions Between Users and the SSFP Computer Systems) flows from a simple division of that interface into three large functional categories:

— The display of information by the system

— The user's manipulation of that information by interacting with the system

— The user's control of specific system functions.

This structure, which is based on the philosophy that the design of the HCI must be shaped by the functions required by the user, results in specific aspects of certain topics appearing in several places in the document. For example, a developer interested in designing a window system would do well to read about windows as features by which information is displayed and features with which the user interacts. In addition, the developer might want to be informed about the various cursor control and selection mechanisms and how they function. Consequently, the developer might want to read three different subparagraphs about windows.

In comparison to a functional organization, a topical organization might contain subparagraphs on windows, menus, the mouse, and so on, in which each (sub)paragraph would contain all of the guidelines on a topic. Such a topical organization would make it difficult for the reader to view the entire system as it functions for the user. In contrast, with the functional organization the reader of the document can easily view the entire system and can, with a little help (described in the next subparagraph), find all of the relevant guidelines for any topic.

Because information on any given HCI topic is distributed in more than one location in the document, the HCIG contains a variety of reader's aids: a table of contents, a subject index, cross references, and a glossary. In addition, paragraphs of the document that have a complex structure contain a graphical overview of the paragraph's major contents. Software developers should pay particular attention to the table of contents, the index, and the graphical overviews to determine how the information required may be obtained from various portions of the guide.

The table of contents lists, to four levels, the main topics and subparagraph headings of the document. These headings are distributed as labels to introduce topics/paragraphs and subtopics/subparagraphs throughout the guide. The table of contents is repeated in graphic (flowchart) form in the reader's guide. Portions of the guide are placed at the beginning of each paragraph so that readers may more easily trace the organization of the paragraph and its relation to information presented previously in the document. Cross references are used primarily in paragraphs 4.1, 4.2, and 4.3 to direct the reader to related guidelines and information.

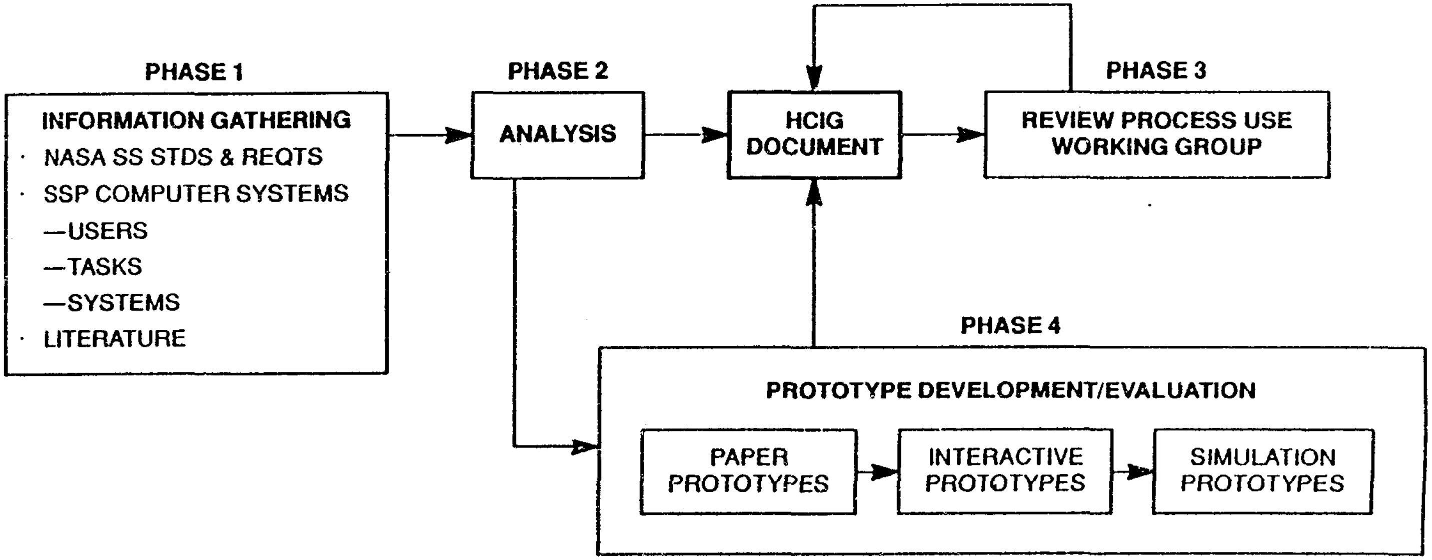


FIGURE 1–1 OVERVIEW OF THE DEVELOPMENT OF THE HUMAN COMPUTER INTERFACE GUIDE

2.0 APPLICABLE DOCUMENTS

The following documents of the date and issue shown include specifications, models, standards, guidelines, handbooks, and other special publications. "Current Issue" is shown in parentheses in place of the specific date and issue when the document is under Level II Space Station Control Board control. The status of documents identified by "Current Issue" may be determined from the Space Station Freedom Program Baseline Activity Index and Status Report.

The documents in this paragraph are applicable to the extent specified herein. The references show where each applicable document is cited in this document.

DOCUMENT NO.	TITLE
SSP 30000	Program Definition And Requirements
(Current Issue)	
References	Paragraphs 1.4, 3.3.1, 3.3.2, and 3.3.4
DOD–HDBK–763	Military Handbook, Human Engineering
February 1987	Procedures Guide
References	Paragraphs 1.3 and 5.0
MIL–STD–1472	Human Engineering Design Criteria for Military Systems, Equipment, and Facilities
References	Paragraph 4.3
NASA–STD–3000, Volume I	
References	Paragraphs 4.2.1 and 5.2.2
NASA–STD–3000,Volume IV	Man–Systems Integration Standards
References	Paragraphs 1.1, 3.3, 4.0, 4.3.1, 4.3.3, and 4.3.3.1
No Number	Subsystem Architectural Control Documents
References	Paragraphs 3.4

2.1 PARENT DOCUMENT

The document in this paragraph is the parent document.

SSP 30534	Software Policies and Information System Standards Document

3.0 SPACE STATION FREEDOM PROGRAM COMPUTER SYSTEMS, USERS, AND TASKS

The design of the Human–Computer Interface (HCI), as exemplified in the guidelines in this document, has been based on an understanding of the Space Station Freedom Program (SSFP) computer systems, the users, and the users' tasks. This paragraph of the document describes information about the systems, users, and tasks that is important in the design of the HCI. Much of the information on users in this paragraph has been summarized from other documentation (ref. 48).

3.1 SPACE STATION FREEDOM PROGRAM INFORMATION SYSTEMS CONCEPT

The Space Station Freedom Program information systems will be developed to support the flight operations of the Space Station and for the transport, processing, distribution, and archiving of payload data. These systems include the flight elements, such as the Data Management System (DMS) and the Communications and Tracking System (C&TS), and the ground elements, such as the Space Station Control Center (SSCC), Software Systems Executive (SSE), Payload Data Support System (PDSS), the Payload Operations Integration Center (POIC), anu the Technical Management Information System (TMIS). Other external elements that are not part of the SSFP but are required for the transport of Space Station information are the NASA Communications Network (NASCOM), the Program Support Communications Network (PSCN), and the Tracking and Data Relay Satellite System (TDRSS).

3.1.1 SSFP INFORMATION SYSTEMS FUNCTIONS

All of the elements listed in paragraph 3.1 will be integrated to perform the following information systems functions when properly integrated.

Additional goals are to do the following:

— Information transfer between flight elements, between ground elements, and between the space and ground elements allowing the operation and support of the Space Station Freedom from distributed ground locations.

— Information transfer to Users, facilitate the open exchange of information between Users, and provide the operations link between the Users and their payloads independent of geographical location.

— Standard, reliable, transparent, information transfer between all elements by utilizing standardized protocols consistent throughout the program life cycle, independent of the user's data format and contents.

3.1.2 SOFTWARE SUPPORT ENVIRONMENT

The primary purpose of SSE is to provide automated rules and tools to minimize the cost and risk associated with Program software development.

Secondary goals are to do the following:

— Provide a common environment for the development and maintenance of all SSFP software. A common environment for software development minimizes the risk involved in a large, extremely complex development effort spread across multiple development centers.

— Provide open access to SSFP software development information which will allow project schedules and status to be tracked at the management level and will allow access to reusable components at the developer level.

— Provide Program–wide enforcement of approved standards and methodologies (e.g., by encapsulating proven methodologies in "smart" tools).

— Minimize cost of software ownership by effective and efficient life–cycle management.

3.1.3 TECHNICAL AND MANAGEMENT INFORMATION SYSTEM

The primary purpose of the TMIS is to provide automated rules and tools to facilitate management of Program development.

Secondary goals are to do the following:

— Maximize the effectiveness and efficiency of technical and management processes over the system life cycle.

— Maximize the effective use of valid system engineering practices over the system life cycle.

— Facilitate the management of information resources.

— Provide technical and management interfaces with SSFP users.

3.2 GROUND–BASED USERS

Ground–based users of the SSFP computer systems may include personnel with a wide variety of abilities and physical capabilities. Accordingly, the user interface should not prohibit use of SSFP by any of these users including the visually, physically, or hearing impaired.

3.2.1 NASA REAL–TIME COMMUNICATIONS NETWORK

TBD

3.2.2 SPACE STATION CONTROL CENTER PERSONNEL

The Space Station Control Center (SSCC) is a Space Station Freedom Program–supplied facility which provides for centralized system management and control for the manned base, including the elements provided by the international partners [European Space Agency (ESA), National Space Development Agency of Japan (NASDA), Canadian Space Agency (CSA)]. Crew and manned base safety are SSCC responsibilities as well. The SSCC provides the system's "templates" for development of Tactical Operations Plans (TOPs), Increment Plans (IPs), and Execute Plans (EPs). It integrates and approves the payload activity schedules developed by the Payload Operations Integration Center (POIC). Crew training facilities are closely associated with the SSCC (and POIC). International partners will support the conduct of operations for their elements by providing responsible flight control staff at the SSCC, as well as providing real–time engineering support from facilities located in their own countries. The SSCC will normally be transparent to the user community during routine payload operations.

3.2.3 PAYLOAD OPERATIONS INTEGRATION CENTER PERSONNEL

The POIC is responsible for integrating and scheduling payload operations. Tactical planning of integration and scheduling will begin two years or more before on–orbit payload operations to determine a payload's placement in the overall space operations schedule. The activity will be similar to Spacelab Mission Planning. The POIC is a Program–supplied facility whose major function is to coordinate user activities for the manned base, building on the template provided by the SSCC. It integrates the user requirements according to user resource envelopes, assists users in periodic "replanning," aids the Interface Working Group (IWG) in user conflict resolution, and supports the various user facilities in real–time or near real–time execution activities. On–orbit crew time and other available resources for users are managed by the POIC in cooperation with the SSCC.

3.2.4 PLATFORM CONTROL CENTER PERSONNEL

TBD

3.2.5 PROGRAM SUPPORT COMMUNICATIONS NETWORK PERSONNEL

TBD

3.2.6 OPERATIONS PLANNING SYSTEM PERSONNEL

The Operations Planning System (OPS) will provide the following three primary functions for planning: manifest analysis, resource distribution, and timelining. Manifest analysis will occur in the tactical planning stage, resource distribution will occur throughout each of the planning stages, and timelining activities will occur during the execute and real–time planning stages. The manifest analysis process involves performing compatibility assessments on selected increment activities. These activities will be grouped and an analysis performed to determine if they are operationally compatible. A short sample timeline of the proposed increment set will be developed to determine if the objectives of the increment are achievable. Activities will be dropped and added until a satisfactory increment set is defined. Resource distribution and assessment includes aligning resource allocations with strategic and tactical guidelines, verifying expected resource availabilities, matching candidate system and payload activities with resources, and identifying resource conflicts. Core system templates will be developed and required resources to support significant Station core activities [Extravehicular Activity (EVA), In Flight Maintenance (IFM), etc.]. These core system resource distributions will be merged with the payload estimates so that an assessment and conflict detection process can begin. The timeline development process provides the capability to schedule increment activities and integrate payload timelines from the core system timelines. The increment activities will be integrated into a timeline and conflict detection and analysis will be performed to determine their compatibility. The required core Station and payload activities will be studied to identify conflicts and problem areas.

3.2.7 FLIGHT DESIGN AND TRAJECTORY PLANNERS

Trajectory planning includes flight and trajectory design and analyses. Flight design analyses and scheduling constraints will be required for all manned and unmanned vehicles that interface with the Freedom Station. Trajectory design will be dependent upon the determination of flight schedules.

3.2.8 FLIGHT CONTROLLER

The current concept of the role of a Space Station Flight Controller is that he or she will function, in general, in a manner analogous to the present Flight Controllers. A Space Shuttle Flight Controller supports the operations of the Orbiter systems in the following ways: prior to a mission, a Flight Controller's activities include integration and documentation related to his or her area of responsibility. During Orbiter missions a Flight Controller is responsible for monitoring, commanding, and controlling one or more Orbiter systems in real time. Specific activities in which the Flight Controllers are involved include:

— providing operations documentation of systems design;

— evaluating systems design and performance;

— developing and verifying software tools for ground support;

— evaluating user requirements and plans as a function of system capabilities;

— providing real–time systems' support;

— monitoring system functioning with an emphasis on predicting, detecting, and isolating malfunctions;

— managing and/or performing system maintenance and reconfiguration; and

— providing resource and command management.

Flight Controllers will have substantial general computer experience as well as training and experience with the SSFP information systems.

3.2.9 PAYLOAD CONTROLLER

The assumption made for this document is that a Payload Controller for the Freedom Station will function, in general, in a manner analogous to the present Payload Controllers. The Payload Controller will handle operational support of both science and non-science payloads, including maintaining the proper functioning of hardware for experiments and experiment support. The Payload Controller will develop procedures for IFM and handling of malfunctions of payloads. The information used by the Payload Controller will include customer data on payload hardware, such as customer schematics and line drawings. The Payload Controller will be experienced with both general computer systems and the SSFP information systems.

3.2.10 PRINCIPAL INVESTIGATOR

The Principal Investigators (PI) for the Freedom Station are anticipated to be similar to those for the National Space Transportation System (NSTS). The PIs will design on–orbit research and other technical investigations or developments, analyze the data from the payload operation, and communicate with the Payload or Station Scientist to perform real–time modifications to procedures. The PIs are likely to have a range of previous experience with computer systems; in addition, they will undoubtedly receive specific training with the SSFP information system but could be expected to be more varied in their level of expertise than the Flight Controllers or Payload Controllers will be. In certain circumstances, the PI may also be an on–orbit user of the SSFP information system; the description of on–orbit PIs is covered in subparagraph 3.3.4, Payload Scientists.

3.2.11 PAYLOAD OPERATIONS DIRECTOR

The Payload Operation Director's function, as in the NSTS, will be to coordinate and take responsibility for scientific payloads. Payload scheduling will be provided by the Mission Planning System (MPS). The Director, with the Payload Controllers and Mission Planner, will also provide operational support for these scientific payloads.

3.2.12 SOFTWARE DEVELOPERS

Software developers (primarily Work Package contractors) will need automated tools to assist with the following:

— requirements analysis

— preliminary and detailed design

— High Order Language (HOL) coding and debugging

— software system construction

— software system testing

- configuration management
- documentation, and
- access to databases of requirements, engineering data and reusable software component libraries.

Since the development sites are distributed across the country, access to remote computers and databases will also be needed.

3.2.13 HARDWARE DEVELOPERS

The hardware developers might be expected to require specialized hardware development tools [such as Computer Aided Design (CAD) and Computer Aided Engineering (CAE)] in the same way as the software development users. The hardware development contractors will have these systems in place in their own facilities. The SSFP information system will provide the means to exchange and access electronic documents and CAD/CAE drawings with remote sites and automated tools to allow their progress to be monitored and assessed by SSFP management.

3.2.14 SYSTEM MISSION PLANNER

TBD

3.2.15 PAYLOAD MISSION PLANNER

The Payload Mission Planner will be responsible for integrating the payload timeline. He/she will utilize the MPS resident at the POIC. The Payload Mission Planner will ensure that all users (payloads) are within their allocation envelopes and the payload plan is free of conflicts. The integrated input will be sent to the OPS at the SSCC for integration into the Station plan.

3.2.16 TESTING AND INTEGRATION PERSONNEL

Testing and integration involves the following: bringing together the hardware and software components that comprise a subsystem or system and verifying that the subsystem or system meets performance and interface requirements.

Activities performed by Testing and Integration personnel through the SSFP information systems will include the following:

- devcloping the plans, schedules, and procedures for integration and testing activities;
- developing the definitions and specifications for the integration and test environment;
- developing simulations (including both hardware and software) to support integration and testing;

- — conducting and monitoring integration and testing activities;
- — documenting plans, schedules, procedures, environments, data, and results from testing and integration; and
- — transfer of plans, procedures, and definitions of hardware and software between tests and test sites. (Ref. 5, section 3.2.2.)

3.2.17 DATABASE ADMINISTRATOR

The database administrator will be responsible for managing data and database systems under configuration control. (Ref. 5, section 3.2.7.)

3.2.18 CLERICAL PERSONNEL

Clerical personnel will perform routine data entry and manipulation tasks. (Ref. 5, section 3.2.7.)

3.2.19 OTHERS

3.3 ON–ORBIT USERS

3.3.1 STATION COMMANDER

One NASA career astronaut, per crew, shall be assigned the duties of Station Commander. The Station Commander's responsibilities as defined in SSP 30000, Program Definition and Requirements Documents, are as follows: ultimate responsibility for crew safety and Space Station Manned Base (SSMB) integrity; principal administration of policy for all on–board personnel affecting crew safety, protocol, and discipline; and coordination of on–board crew activities.

3.3.2 STATION OPERATORS

The Station Operators' responsibilities as defined in SSP 30000 are as follows: act as the shift team specialists for operation, maintenance, repair, and modification of SSMB systems; perform support operations including rendezvous, proximity operations, manipulator operations, free flyer operations, and EVA; and support payload operations as time permits.

3.3.3 STATION SCIENTISTS

The Station Scientists' responsibilities as defined in SSP 30000 are as follows: operate, service, repair, and modify science, application, and commercial payloads; perform support operations including rendezvous, proximity operations, manipulator operations, EVA, and free–flyer operations; and monitor and operate SSMB systems. In addition, one Station Scientist, per crew, shall be designated as the lead for management of on–board user operations support.

3.3.4 PAYLOAD SCIENTISTS

The Payload Scientists' responsibilities as defined in SSP 30000 are as follows: operate, service, repair, and modify assigned science, application, and commercial payloads; and non–safety critical operations of systems to support payload operations.

3.4 DESCRIPTION OF USER'S TASKS

As described in paragraph 1.0, Introduction, the approach taken to designing the user interface for the SSFP computer systems involved identifying, describing, and analyzing the users' tasks, then developing guidelines for these interactions based on the analyses, as well as the scientific literature related to human–computer interactions and accepted, HCI guidelines. This approach is modeled after the traditional approach to HCI design. However, the design for the user interface with the SSFP computer systems differs from that traditional approach in important ways.

Task analyses usually are performed on a limited, but exhaustive, set of well defined tasks. In addition, in the development of most user–system interfaces, the set of tasks is being performed either by users of a version of that system or by users of other closely related systems. For example, in recent redesigns of the FAA Air Traffic Control system, the tasks analyzed in the development of the user interface were tasks currently performed by air traffic controllers (ref. 37). In contrast, an extensive number of diverse tasks are planned to be performed by the users of the SSFP information systems including the SSE, and TMIS, from on–orbit proximity operations to on–Earth software development. Accordingly, the description and subsequent analyses of users' tasks has been based on available operating scenarios (e.g., from subsystem Architectural Control Documents) and on observation of analogous tasks (e.g., Spacelab science payload operations). In addition, the description and analysis of all of the SSFP information systems related tasks would have covered several hundred tasks. To reduce the tasks to a manageable number for detailed analysis, a representative sample of tasks in which space–based and ground–based personnel will interact with the SSFP information systems were selected to be described and analyzed. The criteria for selecting the tasks were as follows:

— the degree of human involvement,

— existing knowledge about the task,

— the similarity to or representativeness for other sets of tasks,

— the task complexity,

— the criticality of the task for the Space Station Freedom Program, and

— the frequency with which the task would be performed.

The definitions of these criteria are provided in Table 3–1, Task Analysis Selection Criteria Definitions. The tasks were selected to maximize the human involvement, existing knowledge, and representativeness, and to represent a range of frequency, criticality, and complexity.

TABLE 3–1 TASK ANALYSIS SELECTION CRITERIA DEFINITIONS

FREQUENCY	The estimated number of times per week that a task will be done on orbit or on the ground.
CRITICALITY	The necessity of the task for the correct and safe operation of the Freedom Station.
COMPLEXITY	The number of steps in the task and the logical relations between steps.
HUMAN INVOLVEMENT	The degree of human involvement including control, management, development, monitoring, and communication.
EXISTING KNOWLEDGE	The availability of simulations, scenarios, or close analogues from STS or other sources.
REPRESENTATIVENESS	The degrees of similarity in function between the task and a number of other tasks.

The tasks selected represented four general types of Space Station Freedom operations: Station Management, Payload Operations, Proximity Operations, and Software Engineering. Specific tasks within each general category were identified as follows:

— STATION MAINTENANCE — Power Management and Distribution (PMAD) (see Table 1, Power Management and Distribution (PMAD) Task Analysis, in Appendix C, Task Analysis Tables), Reboost operation (see Table 2, Reboost Operation Task Description, in Appendix C), and Thermal Control System (TCS);

— PAYLOAD OPERATIONS — Ballistocardiography and Space Telescope;

— PROXIMITY OPERATIONS — Telerobotic/Teleoperator Control;

— SOFTWARE ENGINEERING OPERATION — TBD.

Information from all the task analyses was used in developing the guidelines. (However, Appendix C presents only two task analyses as examples of the outcomes of the task analysis process.)

A task taxonomy, based on the human–computer interface literature (e.g., ref. 44), was developed specifically for describing the SSFP tasks. Table 3–2, Human–Computer Interaction Taxonomy Definitions, shows the task taxonomy used in these analyses. The tasks were described using information from NASA documents (e.g., Architectural Control Document: Data Management System), videotapes of analogous on–ground and on–orbit activities, verbal protocols of simulated tasks, and direct observation of training activities.

Analyses were then performed on each of the described tasks. Input analyses allowed estimation of screen size parameters and general display structure characteristics. Output analyses provided information for anticipating system response speed requirements and feedback considerations.

Link analyses were conducted to determine the connection between user tasks and system commands and operations. The link analyses indicate the frequency with which two subtasks follow in a temporal sequence. The results of the analyses show that no two subtask sequence occurs with great regularity. This finding suggests that (1) the SSFP computer system user interface should be sufficiently flexible to allow nearly any command to be issued from any application and (2) a range of command options should be available to users at any point in a task.

TABLE 3–2 HUMAN–COMPUTER INTERACTION TAXONOMY DEFINITIONS (PAGE 1 OF 3)

MONITOR

— Detect: discover or notice an occurrence (usually unsolicited).

— Search: purposeful exploration or looking for specified item(s).

— Scan: glance over quickly, usually looking for overall patterns or anomalous occurrences (not details).

— Extract: directed, attentive reading, observing, or listening with the purpose of gleaning the meaning or contents thereof.

— Discriminate: roughly classify or differentiate an entity in terms of a gross level grouping or set membership—frequently on the basis of only a limited number of attributes.

— Recognize: specific, positive identification of an entity.

PLAN

— Select: choose an answer, conclusion, or alternative.

— Formulate: generate and put together a set of ideas so as to produce an integrated concept or plan.

— Project/extrapolate: assign an approximate value to a future point based on the value(s) or preceding point(s).

CHECK

— Compare: consider two or more entities in parallel so as to note relative similarities and differences.

— Evaluate: determine the value, amount, or worth of an entity, often on the basis of a standard rating scale or metric.

— Examine: to review closely so as to ascertain the characteristics or nature of an object or occurrence.

SPATIALLY ORIENT

— Locate: determine the relative placement of.

— Align: place one object in a configuration that is compatible with another object such that the two may form an operating component.

— Position: place within specified coordinates.

TABLE 3–2 CONTINUED. HUMAN–COMPUTER INTERACTION TAXONOMY DEFINITIONS (PAGE 2 OF 3)

COMMAND

— Actuate: initiate a process (a set of predefined utilities).

— Direct: provide explicitly authoritative instructions

— Instruct: teach, educate, train, or provide remedial data.

— Adjust: cause small changes to a system or objects so as to maintain stability between it and a larger system.

COMMUNICATE

— Respond: answer or reply in reaction to an input.

— Inform: pass or relay new knowledge or data.

— Receive: get, obtain, or acquire an incoming message.

— Transmit: relay electronically non–mail messages.

— Enter: cause to be registered by the computer system (usually through an ENTER or RETURN key or message).

EDIT

— Copy: designate a portion of an entity and place a duplicate of that portion in a special purpose buffer without removing it from the original entity.

— Cut: remove a designated portion of an entity and place it in a special purpose buffer.

— Delete: remove and destroy a designated portion of an entity.

— Insert: make space for and place an entity at a selected location with the bounds of another such that the latter wholly encompasses the former, and the former becomes an integral component of the latter.

— Filter: selectively eliminate one or more layers of an overlayed composite.

— Aggregate: combine two or more components so as to form a new composite entity.

— Paste: special case of an "insert" operation in which the entity being inserted is copied from a special purpose buffer.

TABLE 3–2 CONTINUED. HUMAN–COMPUTER INTERACTION TAXONOMY DEFINITIONS (PAGE 3 OF 3)

OTHER DATA MANAGEMENT

— Display: reflect an entity on a monitor.

-- Create: initialize and edit a file.

— Open: begin editing a file which already exists.

— Delete/kill: irrevocably destroy a file or process.

— Exit: leave a program/file without deleting it.

— Print: generate a paper printed copy of a file.

— Mail: send messages electronically.

— Save: instruct the computer to record the creation of/changes to a file.

MANUAL TASKS

— Activate: manually engage a system (usually by pressing a button or switch).

— Assemble: construct the components of.

— Stow: store in designated area.

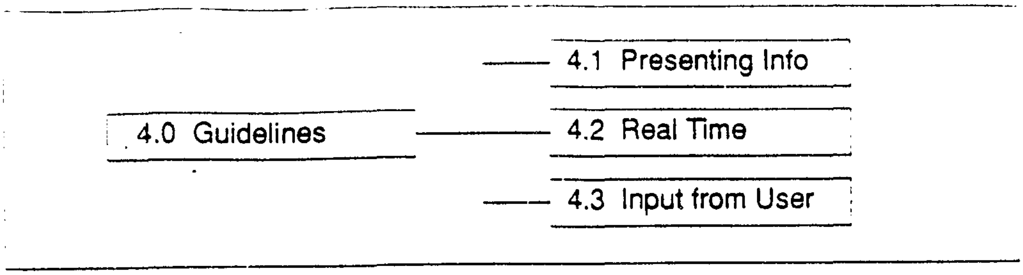
4.0 GUIDELINES FOR INTERACTIONS BETWEEN USERS AND THE SPACE STATION FREEDOM PROGRAM COMPUTER SYSTEMS

The interactions between the various users and the Space Station Freedom Program (SSFP) computer systems can be characterized by three broad categories: the presentation of information to the user; real–time interactions between the user and the system; and the user's input. These categories can be defined, in general terms, as follows:

— The paragraph on the presentation of information to the user addresses issues related solely to the display of information, principally through the visual and/or auditory modalities. This category of guidelines focuses on the processing of the displayed information by the user. Accordingly, this category of guidelines is concerned with such topics as the structures that constitute a display, the organization of those structures (i.e., display syntax), methods of directing the user's attention to specific display areas, and methods of coding the meaning of display elements (i.e., display semantics).

— The paragraph on real–time interactions between users and the SSFP computer systems covers Human–Computer Interfaces (HCIs) that involve a close conceptual and temporal relation between an information display and the user's response, so close that the human–computer interaction cannot be easily classified as involving primarily either information processing or response output. Several of the topics covered under this category previously have been labeled "direct manipulation interfaces." The real–time interactions in this category include methods for moving between displays, windowing techniques, selecting information from a display, user guidance, and interactive dialogue techniques.

— Input from the user is concerned with the entries made by the user into the system to control various system functions. The user provides input to the system through direct control of the cursor or other display positioning cues, alphanumeric characters, and graphic input elements. These directly–controlled elements could also be used by the system to control such functions as payload operations, proximity operations, and subsystem management.

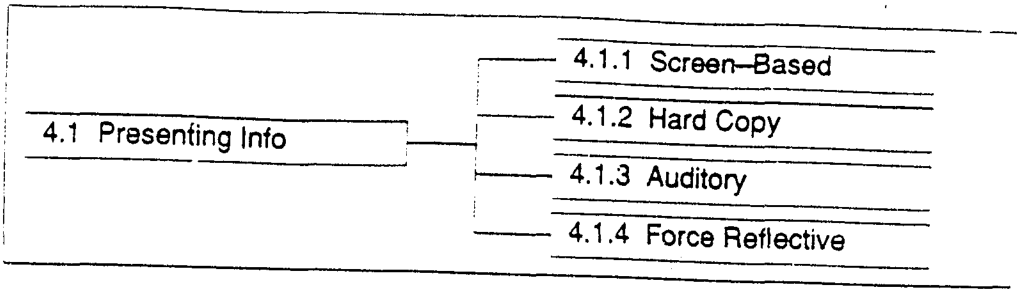


OVERVIEW OF PARAGRAPH 4.0

Due to the dynamics of human–computer interactions, these three categories are not mutually exclusive. However, in developing this document, we have attempted to avoid redundancies by assigning a guideline or related set of guidelines uniquely to a category to the greatest extent possible. But, as paragraph 1.5 described, this organization has resulted in information regarding a given topic appearing in more than one subparagraph/topic. For example, a software developer creating a menu system for an application would obtain menu structure and organization guidelines from paragraph 4.1, guidelines regarding information and menu manipulation from paragraph 4.2, and information regarding the input for menus from paragraph 4.3. Accordingly, the developer might have to consult all three paragraphs. An index and cross–references are provided to aid developers who have to locate information in the various paragraphs of this document.

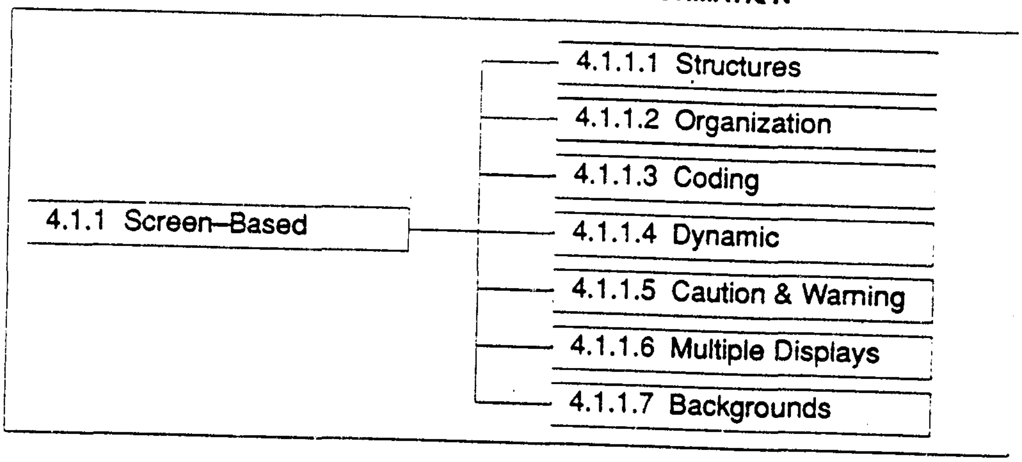
The information in this paragraph of the document falls into three general categories: Man–Systems Integration Standards, paraphrased in this document for easy reference (the reader is also referred to NASA–STD–3000); guidelines based principally on scientific research, industry standards, accepted practices, and/or analyses of Freedom Station tasks; and supporting information, including definitions and descriptions provided within paragraph 4.0 and appropriate rationale.

4.1 PRESENTING INFORMATION TO THE USER



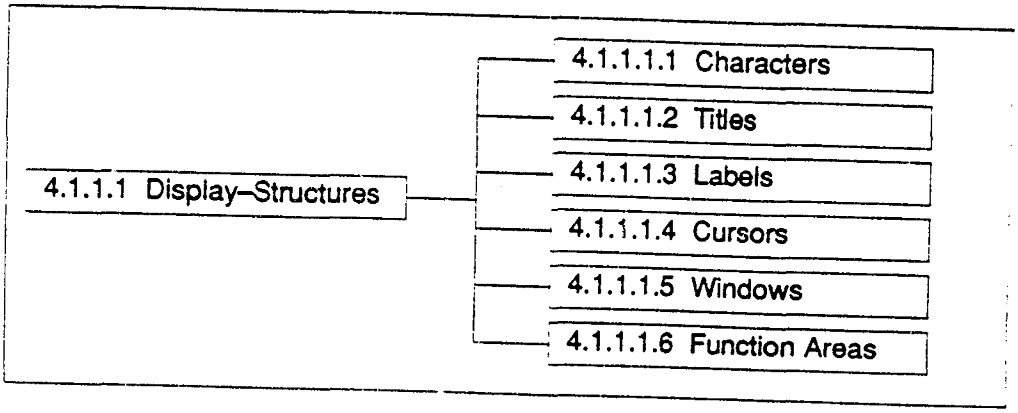
OVERVIEW OF PARAGRAPH 4.1

4.1.1 COMPUTER SCREEN–BASED DISPLAY OF INFORMATION



OVERVIEW OF SUBPARAGRAPH 4.1.1

4.1.1.1 DISPLAY STRUCTURES



OVERVIEW OF SUBPARAGRAPH 4.1.1.1

DEFINITIONS AND DESCRIPTIONS: The term display refers to an integrated, organized set of information required to perform a task or a step in a task. The display might consume an entire screen (the software–controlled visual interface device of a monitor), or cover a well–defined subarea of a screen (e.g., a window). Display structures are information–presenting elements that are consistent in appearance and use across applications. Their functions include providing reference to the user's location, reminding the user what options are available, and providing a visible boundary for user actions.

4.1.1.1.1 CHARACTERS

GUIDELINES:

a. For optimal legibility, the range of character height should be 15.0 to 22.0 minutes of arc with 20.0 minutes of arc being preferred (ref. 3). (See the current release of NASA–STD–3000, Vol. IV, for current SSFP requirements.) Minutes of arc can be converted into height in millimeters as follows:

   Height (in mm)=$2\pi D M_A/21600$

   where M_A is minutes of arc and D is the distance from the user to the screen (in mm).

b. For optimal legibility, characters on a line should be separated by a minimum of one pixel or 20 percent of character width (whichever is greater). (See the current release of NASA–STD–3000, Vol. IV, for current SSFP requirements.)

c. The default value for character width should be 60 percent to 80 percent of the upper case character height. (ref. 7, ref. 82.)

d. Spacing between lines of character groups should be at least one–third the height of the tallest character (ref. 82).

e. The space between the tallest character of a lower line should not be less than one stroke width from a character above it that projects below the line (ref. 82).

4.1.1.1.1.1 CHARACTER FONTS

GUIDELINES:

NASA–STD–3000, Volume IV, provides general Freedom Station font standards for workstations.

a. Characters used on display panels and equipment when viewed under general purpose flood lighting or normal daylight conditions should have a height–to–stroke ratio of 6:1 to 7:1. For example, dark characters on a light background might have a stroke width of 6:1 and light characters on a dark background might have a stroke width of 7:1 (ref. 82). (See the current release of NASA–STD–3000, Vol. IV, for current SSFP requirements.)

b. Simple block styles should be used for labels. Woodson (ref. 82) suggests that the following standard styles are acceptable: Folio Book, News Gothic, Trade Gothic, Future Medium, and Spartan Medium. Gilmore (ref. 3) recommends the following fonts specifically for use on CRTs: NAMEL font for the alphabet, AMEL or AND font for numerals, Leroy font, and Lincoln/MITRE.

c. In general, fonts should have true ascenders and descenders, uniform stroke width, and uniform aspect ratio. Avoid type faces that have extended serifs, internal patterns, or stripes; are italicized, stenciled, shadowed or 3–dimensional; appear like handwritten script or like Old English script; or are distorted to look tall and thin or wide and fat (ref. 82).

d. Under normal illumination (non–dark adapted), dark characters on a light background should be used (ref. 82).

e. In environments requiring dark adaptation, light characters on a dark background should be used (ref. 82).

f. The character should be at least twice as light or dark as the background (ref. 82).

g. For word processing and graphics applications, character size should be under the user's control through selection of font sizes. The font size selection would determine the character height and width, spacing between characters, and the row spacing, in accordance with the preceding guidelines in this section.

4.1.1.1.2 TITLES

GUIDELINES:

a. Each display, including displays contained in a window, should have a unique identifier (title) in a highly visible location that is consistent across all displays (ref. 64, section 4.2.6).

b. Titles should be meaningful and nonarbitrary. Titles should be English words or should use the currently accepted technical term to the greatest extent possible. However, when a display has been reached through a hierarchy of menu selections, the display title may identify the menu selections made (e.g., a string of letters or abbreviations for the menu selections). If a title must be abbreviated, it should be abbreviated according to the guidelines in subparagraph 4.1.1.1.2.2.

c. Titles should be distinguishable from other screen structures and from data (e.g., in all capitals) (ref. 64, section 2.2.8).

RATIONALE: The search time is 13 percent faster for individual words, such as titles and labels, in all capitals (ref. 75).

4.1.1.1.2.1 HIERARCHY OF TITLES

GUIDELINES:

a. Displays may sometimes have several levels of titles (and/or labels). The system should provide visual cues to aid users in distinguishing among the levels in the hierarchy.

b. Character size variation and indentation are two common methods of expressing a hierarchy. Bolding, underlining, and letter case are also frequently used, but the method for their implementation has not been well established. The following serves as an example of how to represent a hierarchy with bolding, case, and underlining:

LEVEL 1 TITLE (LABEL) — Bold, all caps, underlined

LEVEL 2 TITLE (LABEL) — Bold, all caps

Level 3 Title (Label) — Bold, underlined

Level 4 Title (Label) — Bold

LEVEL 5 TITLE (LABEL) — Plain, all caps, underlined

LEVEL 6 TITLE (LABEL) — Plain, all caps

Level 7 Title (Label) — Plain, underlined

Level 8 Title (Label) — Plain

4.1.1.1.2.2 ABBREVIATIONS AND ACRONYMS IN TITLES

DEFINITIONS AND DESCRIPTIONS: The guidelines in this subparagraph should be applied to all textual elements of displays that might use abbreviations and acronyms, including labels, headings, messages, menus, running text, data forms, and tables.

GUIDELINES:

a. Abbreviations and acronyms should be used only if a display does not have sufficient space for the unabbreviated word or if the abbreviation or acronym is more frequently used than the full word or phrase (e.g., NASA).

b. Abbreviations and acronyms should be used only if significantly shorter than the complete word, saves needed space, and can be understood by the user population. (ref. 64, section 2.0.16.)

c. When abbreviations are used, commonly recognized abbreviations should be chosen (i.e., NASA's conventional abbreviations).

d. If a descriptor must be abbreviated, and conventional abbreviation does not exist, then, to the greatest extent possible, follow a single, simple rule to generate the abbreviation. The same rule should be applied throughout the SSFP computer system user interface for creating all new abbreviations. Abbreviation by truncation is often the best choice (ref. 64, section 2.0.18). For example, the word abbreviation might be truncated as abbrev. (If abbreviation by truncation is ineffective, for example, if the word ending is particularly informative, consider systematically removing the vowels to abbreviate.)

e. When acronyms are used, NASA–recognized acronyms should be chosen; that is, new acronyms should not be coined for the purpose of the user interface.

f. When abbreviations and acronyms are used, the abbreviation or acronym and the associated word or phrase should be easily available to the user (e.g., through a Glossary contained in on–line Help).

4.1.1.1.3 LABELS

DEFINITIONS AND DESCRIPTIONS: A LABEL is a descriptor that identifies all associated data items, either for an individual item or for a group (e.g., a column heading).

GUIDELINES:

a. The referent for a label should be consistent across and within displays (ref. 64, section 2.3.7).

b. Labels should be meaningful. Labels should be English words or should use the currently accepted technical term.

c. Labels should consist of the entire word or sequence of words, rather than an abbreviation, whenever space permits. If a label must be abbreviated, it should be abbreviated according to the guidelines in subparagraph 4.1.1.1.2.2.

d. Labels should not include contractions, short forms, or punctuation unless absolutely necessary for meaning, to accommodate space limitations, or unless the label is an accepted standard.

e. Labels should include the unit of measure for the data described (ref. 64, section 2.2.10).

f. To the greatest extent possible, labels should maintain a consistent location between displays (ref. 64, section 2.2.7)

g. Labels should be centered and in close proximity above columns in a numerical data matrix.

h. Labels should be centered and in close proximity to rows in a numerical data matrix (ref. 64, section 2.2.9). Because people read English text from left to right, labels should generally be placed to the left of the row whenever possible.

i. Labels should be left justified in a tabular array with numerous subheadings or sublabels.

j. Labels should be differentiated (e.g., underlined and in all capitals) from other screen structures and data in a unique and consistent manner (ref. 64, section 2.2.8).

RATIONALE: The search time is 13 percent faster for individual words, such as labels, in all capitals (ref. 75).

4.1.1.1.4 POSITION DESIGNATION (CURSORS)

DEFINITIONS AND DESCRIPTIONS: A CURSOR is a display structure that is used to indicate the position of the user's operation on the display. Cursors serve the following two different functions: placeholding and pointing. Placeholding involves showing the location of the immediately previous operation or the point at which the user has moved the cursor. User operations marked by a placeholder cursor include word processing, data manipulation, (e.g., in spreadsheets or data forms), data entry, and graphics, in which the cursor indicates the location of the next character or graphical operation. Pointing involves indicating the user's position in relation to certain other display structures such as icons, menu bars and items, and scroll bars. The pointing cursor can be used to position the placeholder cursor. See Figure 4–1, Examples of Cursors, for examples of cursor styles.

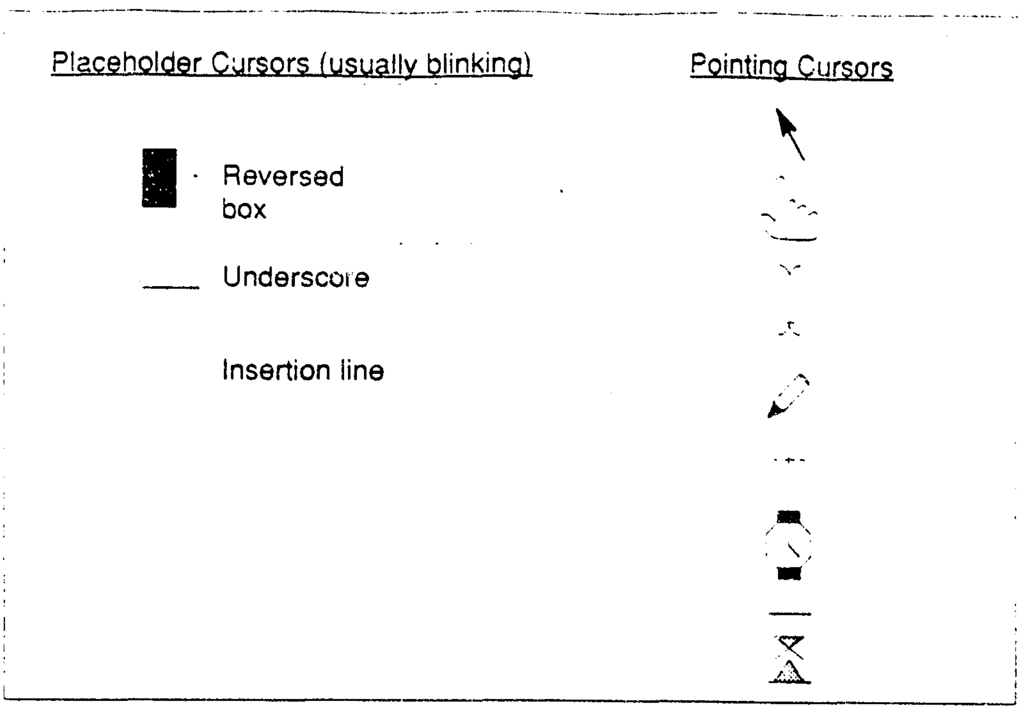


FIGURE 4–1 EXAMPLES OF CURSORS

GUIDELINES:

a. All cursors should be distinctive against all backgrounds and should be easy to locate.

b. The shapes used for all cursors should be unique with respect to all other display structures and should be consistent within and across applications or functions.

c. Cursors of different shapes should be used for different purposes. The shape of a cursor should reflect the state of the system or processing mode. A specific cursor should be uniquely assigned to a specific purpose to provide state or mode information to the user. For example, a straight line cursor might be used as the placeholder cursor to indicate entry position in a word processing task, an arrow might be used as a pointing cursor to indicate screen structures, and an X–shaped pointing cursor might be used when the user cannot interact with the system. Figure 4–1 shows examples of different cursors. Within this general framework, the number of cursor shapes used should be kept to a minimum.

4.1.1.1.4.1 POINTING CURSORS

GUIDELINES:

a. The pointing cursor should be visible to the user at all times and may obscure characters unless it interferes with performance within an application. Pointing cursor quality may be maintained by assigning image quality priority to the cursor; to maintain pointing cursor quality, the cursor should obscure other characters, not vice versa.

b. Pointing cursors should maintain image quality throughout an entire range of motion within the display. The position of the pointing cursor should be clearly visible during movement from one screen position to another. Flicker should be minimized.

c. The pointing cursor should not blink.

d. The pointing cursor should maintain its size across all screen and display locations.

e. The movement of the pointing cursor should be related directly to the movement of the cursor control device (see paragraph 4.3). The movement of the pointing cursor should appear to the user to be smooth and continuous with smooth and continuous movement of the cursor control device. The pointing cursor should not move in the absence of any input from the user.

f. To the greatest degree possible, pointing cursors should be completely graphic and should not contain a label. However, if a pointing cursor includes a label, the text should be large enough to be readable.

g. When there are multiple cursor control devices (e.g., both a mouse and a trackball), a unique pointing cursor shape should be associated with each device.

4.1.1.1.4.2 PLACEHOLDER CURSORS

GUIDELINES:

a. The placeholder cursor should only be visible when text entry is possible. The placeholder cursor may not obscure characters.

b. There should be only one placeholder cursor per window.

c. The placeholder cursor should assume the height and/or width of the text characters adjacent to it.

d. If placeholder cursor blinking is to be used to direct the user's attention, the default blink rate should be 3 Hz (ref. 14). A blinking cursor need not obscure characters; for example, the blinking cursor may be an underline that does not cover the entire character.

e. At the initiation of a task, an application, or a new display, the user should be able to determine the location of the placeholder cursor without an extensive search. For example, the cursor might be placed initially at the first data field in a dataform, at the upper left corner of a blank display in a word processing task, and immediately following the last character of a word processing display containing alphanumeric characters. Following the initial placement of the placeholder cursor, the position of the cursor should also be under the user's control.

4.1.1.1.5 WINDOWS

DEFINITIONS AND DESCRIPTIONS: A WINDOW is a "subdivision of the display screen where one set of output is displayed" (ref. 42, p. 497). Examples of windows can be seen in Figures 6–3, Pop–over Notes Window; 6–4, Pop–over Help Window; and 6–5, Initial Reboost Display.

GUIDELINES:

a. The default width for a generic text window should cover from 67 percent to 100 percent of the full screen.

RATIONALE: Duchnicky and Kolers (ref. 20) found that when users read continuously scrolling text (at a rate set by the user), line lengths of 52 to 78 characters provided the fastest performance.

b. The default size for text windows should be at least four to seven lines of information. Beyond four to seven lines of information, the default window size should be a function of the amount of information to be displayed in the window.

RATIONALE: Duchnicky and Kolers found that window sizes of four lines provided better performance than those with fewer than four lines. Windows with 20 lines showed little advantage over windows with four lines (ref. 20). Other research has shown that search time is slower in a one–line window than in the next largest size (seven lines), but did not vary appreciably among 7, 13, and 19 line windows (ref. 21).

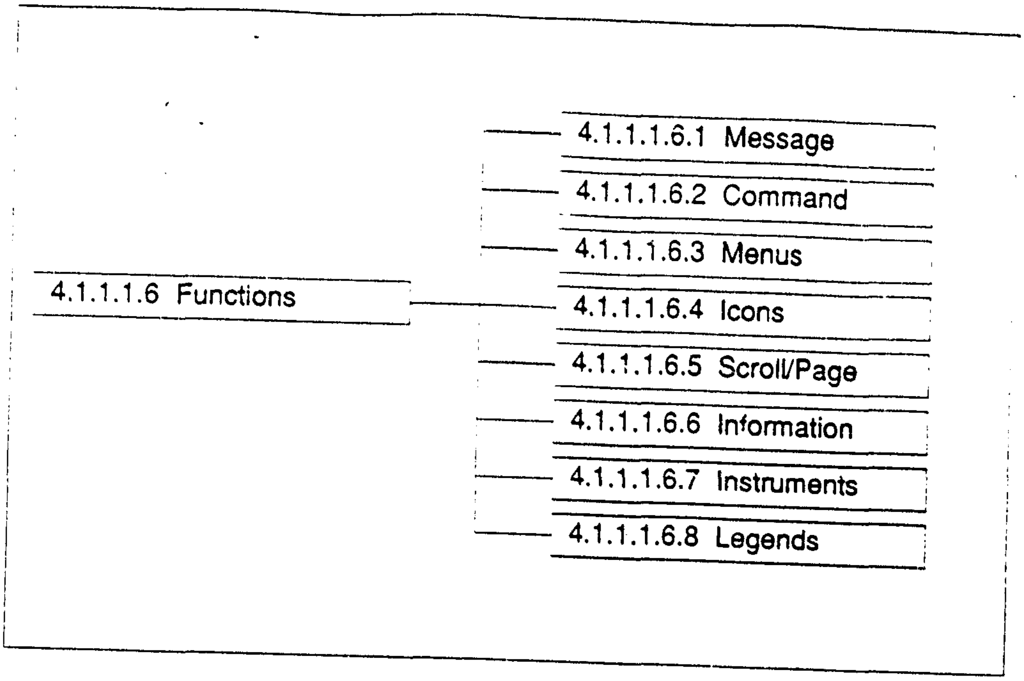
c. The dimensions of the window should be under the user's control (see subparagraph 4.2.3 for guidelines on user interactions with windows).

d. The user should have the ability to scroll through the contents of a window both horizontally and vertically, if scrolling is required at any point in an application (see subparagraph 4.1.1.1.6.5 for detailed guidelines related to scrolling structures).

e. Windows should have a rectangular shape. The window should be framed by a border of a single line. The frame should expand and contract with the window.

f. The title of a window should be positioned in a consistent and highly visible place (e.g., centered at the top of the window). The title should accurately and uniquely describe the contents of the window (see the guidelines on titles in subparagraph 4.1.1.1.2).

4.1.1.1.6 FUNCTION AREAS



OVERVIEW OF SUBPARAGRAPH 4.1.1.1.6

DEFINITIONS AND DESCRIPTIONS: FUNCTION AREAS are specific locations that are reserved for a specific purpose. Function areas can occur anywhere on the screen; that is, on the primary display or within a window which is part of the primary display. A function area would be capable of appearing on either all displays or all displays within a category. Whether the function area appeared on the display might be under the control of the user. Examples of function areas include areas reserved for the date and time, a menu bar, a scroll bar, a paging area, and a command line.

4.1.1.1.6.1 MESSAGE AREAS

DEFINITIONS AND DESCRIPTIONS: A MESSAGE AREA is a specialized function area for text communication from another user at a different workstation or delivered automatically by the system to describe a system state or operation (e.g., a status message). The message area will not display emergency or critical messages. All Caution and Warning (C&W) messages will be displayed in a separate area reserved for that function. See subparagraph 4.1.1.5 for guidelines concerning C&W messages.

GUIDELINES:

a. Messages should be coded according to their priority so that the user is aware of the timeliness in which he or she needs to respond (ref. 64, section 5.5.5). (See the current release of NASA–STD–3000, Vol. IV, for current SSFP requirements regarding alarm classification and annunciation requirements and NASA–STD–3000, Vol. I, for current SSFP requirements regarding the alarm classification system.)

b. Message areas should be displayed in a window. The windows used for messages from the system and messages from other users should be located at unique, consistent locations. However, once the window appears on the display, it should be under the user's control. The user should be able to remove the message area from the screen, revive it, move the window within the screen, and change the window's size. Each of these actions should require only a single action by the user.

c. Visibly and spatially distinct areas should display messages from the system and messages from other users. However, the default sizes and the capabilities of the two message areas should be similar, except as indicated in specific guidelines below.

d. By default, the width of message areas should extend across the entire display width and the default height should be three lines (ref. 3, p. 17).

e. Each time a new message appears, the default window should be displayed although the message areas' height and width should be adjustable by the user. (Note: C&W messages that may need to be seen in their entirety are discussed in subparagraph 4.1.1.5 of this document.)

f. The user should be informed when a message extends beyond the area that the window is able to display.

g. Messages should be stored in a message queue (ref. 64, section 5.5.2) that is available to the user. For example, the user might be able to scroll through a log file containing the message and the time, date, and origin of the message.

h. For real–time operations, messages should be time–stamped.

i. Notification of messages received should be provided automatically at log on and while the user is logged on (ref. 64, section 5.5.4).

j. Notification of incoming messages while the user is logged on generally should be nondisruptive (ref. 64, section 5.5.5).

k. Notification of incoming messages should not interrupt the user's current task and should not automatically overwrite the screen areas where the user is working. For example, the system might indicate message arrival to the user by an advisory notice in a portion of the display reserved for that purpose.

l. Explicit user actions should be required to access incoming messages.

m. The user should be able to rearrange messages such that they can be reviewed regardless of the order in which they are queued (ref. 64, section 5.5.13).

n. Users should be able to view, edit, send, and delete messages. The default condition should be to view messages.

4.1.1.1.6.2 COMMAND AREA

DEFINITIONS AND DESCRIPTIONS: The COMMAND AREA is the screen location where the user can input the User Interface Language (UIL) (see subparagraph 4.2.8.1.1.1).

GUIDELINES:

a. A single command area should be in a consistent location on the screen.

RATIONALE: A user can only provide input into one command area at a time, even if the system in which the user is working permits multitasking. The presence of multiple command areas (e.g., several windows whose sole function would be for the user to input UIL) would increase the user's workload without a commensurate increase in functionality.

b. The command area should be visibly distinct from all other screen structures.

c. The command area should be contained in a window that the user can open, close, or resize.

d. The user should always have ready access to the command area. For example, the command area might be in a window that could not be covered by other windows or could easily be "popped" to the front of all other windows by selecting a software–based button.

e. When the command area window is first opened, the placeholder cursor should be in the leftmost position on the first line. At other times, the user should have the ability to place the placeholder cursor at any location in the command area.

f. In general, opening the command area window should not interfere with the user's ability to view other display structures. However, when the command window is revived and contains commands, the command area should be the same in size, content, and cursor placement as when it was removed.

g. When the command area is opened, the user should immediately be able to enter a new command for execution, access previously entered commands and execute them as they appear, or modify previously entered commands and execute them. The user should be able to edit, cut, paste, and re–execute these commands by the same methods as are used in text input and editing (see guidelines in subparagraph 4.2.4.1).

RATIONALE: Making previous commands available will provide continuity between command input episodes and may reduce time in repetitive tasks.

4.1.1.1.6.3 MENUS

DEFINITIONS AND DESCRIPTIONS: MENUS are specialized function areas that display categories of user response alternatives. The successful use of menus requires recognition memory rather than recall memory and thus provides for fewer errors and less confusion. Menus are especially beneficial to novice or infrequent users. In addition to providing for selection among several choices, menus allow for the entry of default values or parameters as well as the toggling of multiple options.

There are two basic classes of menus: menus as permanent display structures (i.e., menus which cannot be moved or removed) and menus as hidden display structures, in which visibility of the menu or part of the menu is by user request only.

Menu components include a title or category label area and a group of items in which one or more of the items is selectable by means of a pointing device selection action or keyboard entry. The guidelines in this subparagraph describe menus as display structures only. For more detailed guidelines on menus as a dialogue technique, see subparagraph 4.2.8.1.2.

GUIDELINES:

(NOTE: These guidelines apply to permanent and user requested menus for a display which covers the entire screen as well as those contained in individual windows.)

a. In general menus should be implemented in a list format, displaying each option on a new line.

RATIONALE: This format aids scanning and provides for quick responses (ref. 64, section 3.1.3.3).

b. Menu items should be placed in a menu list in a manner which reflects frequency of use as well as functionality. Menu items of similar functionality should be grouped together. Within a functional grouping, the most frequently used menu items should be the easiest for the user to access. If there is an order relationship among the tasks or steps in a task that the menu items represent, the items should be displayed in that order.

c. Menu titles/category labels should be brief, descriptive of the contents of the menu, and distinctive from other menu titles/category labels. Menu titles/category labels should be consistent with corresponding functions across applications (e.g., the label "Edit" for editing functions in text, graphics, and tabular applications).

d. If menu selections have equivalent representations in the UIL, the wording and organization of the menu should be consistent with the UIL wording and structure.

e. All menu items should be visible to the user without scrolling. This guideline applies to permanent menus, as well as for pop–up or pull–down menus when they are popped–up or pulled–down, respectively.

4.1.1.1.6.3.1 MENUS AS PERMANENT DISPLAY STRUCTURES

DEFINITIONS AND DESCRIPTIONS: PERMANENT MENUS are menus which are constantly visible and are an inseparable part of the display. These menus cannot be removed or hidden unless the entire window is closed or the display itself changes. Menu items can be chosen by selection with a pointing device or the input of an item code (e.g., number, letter).

GUIDELINES:

a. Permanent menus may be used in cases where there is no pointing device available (although their use is not restricted to this case). Consequently, some type of code (number or letter) is usually entered through the keyboard to indicate a menu item choice. There should be one standard design for the input prompt that is used across all applications, for example: "ENTER CHOICE: __". There should be a text prompt delimiter (e.g., a colon) as well as an underscored area representing the maximum input length.

b. The location of the input prompt should be the same on all displays. For example, locating the selection code prompt near the bottom of the screen may be beneficial.

RATIONALE: This minimizes the head/eye movement when the user is locating the appropriate key (ref. 64, section 3.1.3.8).

4.1.1.1.6.3.1.1 HIERARCHICAL BRANCHING WITH PERMANENT MENUS

DEFINITIONS AND DESCRIPTIONS: HIERARCHICAL BRANCHING is a method of structuring menu items that are hierarchically related. When implemented with permanent menus, moving to a subordinate menu level requires that an entirely new display be presented.

GUIDELINES:

a. If hierarchical branching is used, each subordinate menu should be visually distinct from each previous superordinate menu. Examples include the display of level numbers, a graphical stacking effect, etc.

RATIONALE: Successful user operations depend on a knowledge of context. The user needs to know the levels from which the current display menu came and how far down in the hierarchy the current menu is.

4.1.1.1.6.3.2 MENUS AVAILABLE AT USER REQUEST

DEFINITIONS AND DESCRIPTIONS: MENUS AVAILABLE AT USER REQUEST include pull–down and pop–up menus activated by a pointing device selection action. A currently popular way of accessing hidden menus is by means of a pointing device entry in a menu bar area.

A MENU BAR is a specialized function area that displays categories of user response alternatives. Items within a category are displayed from the menu bar only when the user activates the category (e.g., by a selection action). For an example of a menu bar, see Figure 5–1. Another method makes the appropriate menu available by a pointing device entry at any location on the display, where the location of the selection action determines the menu contents. Alternatively, the menu bar could appear in some form at the selection action location. This would prevent the large cursor travel distance associated with menu bars on large displays (e.g., 19 inch) and eliminate the necessity for an arbitrary function/location association, but would provide the user with the memory aid of the menu categories. In such an implementation, the menu bar would return to its permanent location at the top of the display as the selected operation is carried out.

PULL–DOWN MENUS: The menu items in a pull–down menu are normally "hidden" from the users view and accessed by the user holding the selection button down over the desired menu bar label. Selection of the text label activates the presentation of a list of menu items which are attached to the menu bar giving the user the impression that the list of items was pulled down from the menu bar. While the selection button is down, the user can move the cursor over the selections and release the selection button over the desired menu item. This menu is only visible to the user as long as the selection button remains depressed.

POP–UP MENUS: Pop–up menus are very similar in appearance and function to pull–down menus with one exception. Pop–up menus are generally activated or brought into full view by a complete selection action; for example, pressing, then releasing a selection button. Menu items are selected by a selection action on the desired menu entry. Pop–up menus remain visible until another user action takes place to hide the menu or make a selection. If the user wants to hide the menu without making a selection, there is generally a close box or "exit menu" item available.

GUIDELINES:

a. The height of a menu bar should be sufficient to contain standard text characters which serve as menu category labels, as well as space above and below the text characters.

b. Category labels on menu bars should be centered in the vertical dimension. Horizontally, category labels on the menu bar should be separated by enough space to be distinguishable as separate items, i.e., by at least two standard character widths.

RATIONALE: One standard character width would be required to separate adjacent words in a multiword category. To indicate separate categories, more than one width would be needed.

c. Category labels on the menu bar should be brief, descriptive of the contents of the menu, and distinctive from other category labels.

d. The categories listed across the menu bar should be organized systematically. For example, the categories on the left side of the menu bar might be system functions that apply across all (or most) applications. The categories on the right side of the menu bar might be those that are specific to the currently–active application. Within this general spatial layout, both the system–wide and specific categories would be ordered from left (the category containing the most frequently used actions) to right (the category containing the least frequently used).

e. The number of categories listed on the menu bar should not exceed the length of the bar. That is to say, reading the menu bar should not require scrolling.

f. The category label for a set of menu items should be visually distinctive if none of the menu items is currently selectable.

g. Menu bars should be placed at a consistent location in all displays, for example, at the top of each display.

4.1.1.1.6.3.2.1 HIERARCHICAL BRANCHING WITH USER–REQUESTED MENUS

DEFINITIONS AND DESCRIPTIONS: When implemented with user–requested menus. HIERARCHICAL BRANCHING provides for selection among alternatives without requiring the opening and closing of a series of menus; thus, the entire hierarchy is contained in one menu.

GUIDELINES:

a. The number of levels available via menu item selection should be minimized.

b. When menu items within a list are actually branch points to lower level selections, access to lower levels should be achieved by the simultaneous presentation of the superordinate and subordinate lists. This presentation of all menu lists prevents the user from forgetting which superordinate selections were made.

4.1.1.1.6.3.3 CHOOSING MENU TYPES

GUIDELINES:

a. The use of permanent menus should be minimized because they require dedicated display space and more paging activity (because the application must return the user to the main menu page at every task change). However, permanent menus might be used whenever it is beneficial to examine every option in detail or when the amount of text in each menu item is large.

b. User requested menus should be used whenever possible. The savings in display space is substantial. Among the types of user–requested menus, pull–down menus provide the following two advantages over pop–up menus: the menu bar serves as a useful mnemonic aid, showing the user the command categories available in the menu; and gaining visual access to the menu items within a category, selecting the item, and removing the menu can be accomplished with a minimal number of actions. The primary advantage of a pop–up menu over a pull–down menu is that, depending on the specific implementations, the user may have immediate access to the menu at the screen location of the selection action. The ideal user–requested menu design would provide the user with a reminder of the menu categories and would allow the user to select an item with few actions and little movement of a cursor on the screen.

4.1.1.1.6.4 ICONS

DEFINITIONS AND DESCRIPTIONS: ICONS are pictorial representations of objects or actions. Icons can be used simply as a symbol of an object or action that cannot be manipulated, or as a symbol on which the user can directly act (e.g., selecting the icon to open or move applications or files) (see subparagraphs 4.2.8.1.5 and 4.2.8.3.6 of this document for guidelines related to interactions with icons).

GUIDELINES:

a. The primary use of icons in graphic displays should be to represent concrete objects or actions..

b. The object or action that the icon represents should be visible in the icon. This guideline has the following two implications: the size of the icon should be large enough for the user to perceive the representation and discriminate it from other icons, and the representation should be pictographic whenever possible.

c. The external geometry of the icon should be the informative feature for the user (ref. 34). Accordingly, an icon should not consist of a pictograph contained inside of a generic geometric figure that has no information value (e.g., circle, square, or other border shapes).

RATIONALE: Icons that are bordered by irrelevant shapes for the sake of aesthetics are actually made more visually complex. The additional irrelevant visual information interferes with the processing of the relevant information (the icon itself) (ref. 45).

d. Icons should be simple, closed figures when possible.

RATIONALE: When icons are too visually complex, they are not quickly recognized. This eliminates the primary advantage of using icons: quick recognition. Simple, closed figures are processed more efficiently than are open figures (ref. 66).

e. To the greatest extent possible, icons should be accompanied by a text label, especially when the icons do not closely resemble the symbolized object or action (ref. 13).

f. To the extent that it does not clutter or cause distortion of the icon, icon labels should be incorporated into the icon itself (ref. 69).

RATIONALE: When icons are designed such that the label is inside of the icon, the number of perceptual objects is reduced, resulting in enhanced processing of the label and the icon.

g. Under default conditions, icons should be grouped spatially together on the display. The user should be able to change the spatial location of an icon.

4.1.1.1.6.5 DISPLAY STRUCTURES FOR SCROLLING AND PAGING

DEFINITIONS AND DESCRIPTIONS: A display structure for scrolling and paging permits the user to move either horizontally or vertically through a display or connected sequence of displays. Scrolling provides the appearance of continuous movement, whereas paging provides movement in discrete steps. Figures 6–2 through 6–21 show examples of one implementation of a scrolling/paging structure, the scroll bar. Note that some scroll bars are located on the primary display and some on windows in those figures. Other possible implementations include scroll arrows, a scroll wheel, a pictograph of a scroll, or a pictograph of a page.

GUIDELINES:

(NOTE: These guidelines apply to both displays which fill the entire screen and displays contained in individual windows.)

a. The display structure used for scrolling and paging should be common for all files through which the user might need to move in a continuous or discrete fashion, including text files, data forms, and graphics files (see also subparagraph 4.2.2.1).

b. Structures for horizontal scrolling/paging should appear only on displays for which horizontal movement is appropriate. Similarly, structures for vertical scrolling/paging should appear only on displays for which vertical movement is applicable.

c. Only one scrolling/paging structure should be used for vertical movement in a display and one for horizontal movement in a display. The placement of the scrolling/paging structures should clearly indicate their function for vertical or horizontal movement. For example, one scroll bar might be placed along one of the side borders of the display for vertical scrolling and another scroll bar might be placed along the top or bottom (opposite the menu bar) of the display for horizontal scrolling.

d. A scrolling/paging structure should indicate both the absolute and relative positions of the user in the data file. For example, a page icon on the scroll bar might indicate the absolute position by containing the page number in the data file and indicate the relative position by means of the spatial location of the icon on the scroll bar.

e. The function of the scrolling/paging structure should be clearly indicated by either a textual or graphic label. For example, a graphic label for the scroll bar might be a scroll icon (see Figures 6–2 through 6–21 for the typical arrow scroll icons).

4.1.1.1.6.6 INFORMATION AREA

DEFINITIONS AND DESCRIPTIONS: The INFORMATION AREA contains general purpose information that would be helpful to all users, including the time, date, and version number of any application.

GUIDELINES:

a. The information area should display the appropriately formatted time information necessary for a user's task and location. For example, Freedom Station crewmembers, users who communicate directly with the crew [e.g., Flight Controllers from the Space Station Control Center (SSCC)], and science users with payloads on board the Station would need to have a common time system. The common time might be Greenwich Mean Time or SSCC time.

b. Date and Time information should be shown in a consistent location.

c. In addition, the information area might also display the local time for ground–based users.

d. Examples of Date and Time areas can be seen in all paragraph 6.0 Figures.

e. Users should have the capability to access and permanently display the version number of the current application.

f. Users should be able to remove any information in this area from the display, either for a single session (e.g., by removing the window in which the information is contained) or for all sessions (e.g., by a change in their user profile).

4.1.1.1.6.7 SOFTWARE–BASED INSTRUMENT PANELS

GUIDELINES:

a. Software–based displays of instruments should be displayable in a function area. The panel should be contained in a labelled window and should be closely analogous to the hardware panel that is represented. Specific instruments on the panel should be clearly visible and should be labelled appropriately, following standards for hardware panel labels. (See the current release of NASA–STD–3000, Vol. IV, for current SSFP requirements on panel labels.)

b. This function area should be displayed only after a specific request from the user.

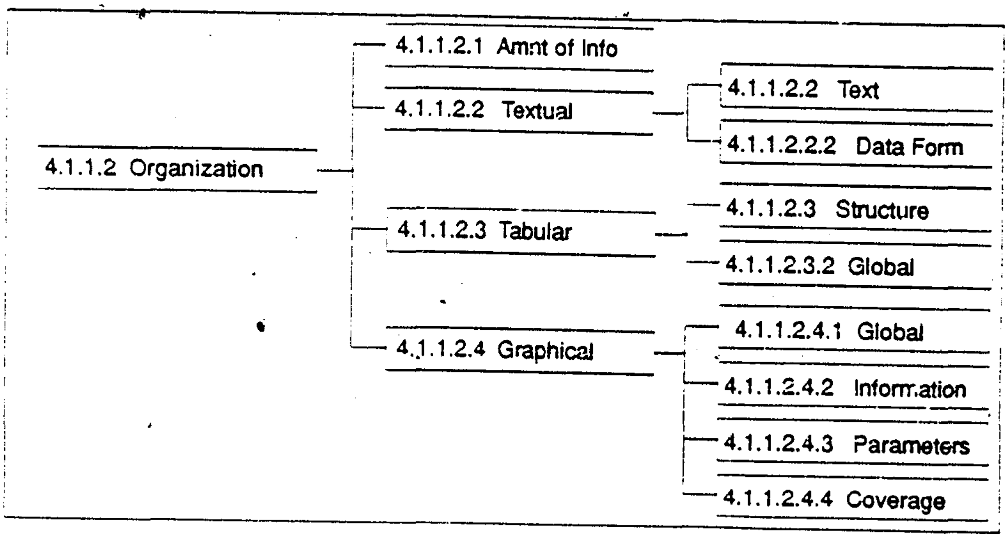
4.1.1.1.6.8 LEGENDS

GUIDELINES:

a. A function area should display legends used to relate symbols to their referents, especially for graphical displays (e.g., data graphs, maps, and schematics).

b. The legend area should be in a consistent location on all graphic displays.

4.1.1.2 ORGANIZATION OF INFORMATION IN A DISPLAY



OVERVIEW OF SUBPARAGRAPH 4.1.1.2

DEFINITIONS AND DESCRIPTIONS: The previous subparagraph described many of the elements of an HCI that carry information. Organizing that information in a display is one of the most important aspects of the HCI. Good organization permits the user to read information from the screen rapidly and accurately. The performance of the users and ultimately of the system depends on rapid and accurate processing of information.

4.1.1.2.1 AMOUNT OF INFORMATION TO PRESENT ON DISPLAYS

DEFINITIONS AND DESCRIPTIONS: The amount of information to present refers to the display area that should contain information or, when this is possible to define, the percentage of screen area to be filled with information. However, the arrangement of information that is presented is more important than the amount of the information: Good organization allows the user to use more of the displayed information.

GUIDELINES:

a. See the current release of NASA–STD–3000, Vol. IV, for current SSFP requirements on the density of display information.

b. To the greatest extent possible, each display should contain all of the relevant information for the user's task or current step in a task but only the relevant information.

RATIONALE: Failure to provide all of the information needed will cause errors or will force the user to spend time accessing the additional needed information. Providing unneeded information increases the number of errors and the time required to do a task (e.g., ref. 19, ref. 23, ref. 33, ref. 63, ref. 69).

c. A text display should show at least four lines of text when simple text is displayed. (ref. 64, section 2.1.)

d. In a tabular display, no more than 40 groups of data should be displayed simultaneously, where each group subtends five degrees of visual angle or fewer (ref. 72). If more than 40 groups were required to display data in a table, the groups should be assigned to separate but related displays (e.g., in multiple layered windows; see subparagraph 4.2.3.3).

RATIONALE: An inverse relation exists between the number and size of groups of data. Approximately 20 groups subtending five degrees of visual angle can be displayed on a standard screen. As group number approaches the recommended upper limit, the average group size will decrease.

e. When possible, graphical displays should show a sufficiently large amount of the graphical object(s) so that the user can identify the object(s). The term GRAPHICAL OBJECTS is used here to mean the graphically–displayed information of primary interest to the user (e.g., a data graph or a schematic diagram). When a graphical object is too large to show on an entire display, the SSFP computer system should provide a secondary display that shows the entire object and indicates the area of the object that is currently in the primary display. For example, the secondary display might be an insert within the primary display or might be a separate display that the user can access.

f. The following general guidelines for the display areas of individual screen structures in textual displays are provided in the various subparagraphs that describe those structures:

Message area — see subparagraph 4.1.1.1.6.1

Command area — see subparagraph 4.1.1.1.6.2

Menus — see subparagraph 4.1.1.1.6.3

Scrolling and Paging — see subparagraph 4.1.1.1.6.5

Information area — see subparagraph 4.1.1.1.6.6

g. A user should be able to easily transfer data from one application to another (e.g., experimental data to graphics or to a word processor).

4.1.1.2.2 FORMAT OF TEXTUAL DISPLAYS

DEFINITIONS AND DESCRIPTIONS: The primary elements of textual displays are characters and a cursor, as well as general display structures present with most applications (e.g., a menu bar or a message area). The major focuses of this subparagraph are the organization of information on a textual display and the placement of display structures.

4.1.1.2.2.1 TEXT

DEFINITIONS AND DESCRIPTIONS: TEXT consists of alphanumeric character strings in linear arrays, making up words, sentences, and paragraphs. Text on a video screen is similar to that in a book. However, under many conditions, reading text on a CRT is significantly slower than hard copy reading (ref. 41). However, comprehension of text is not necessarily different between a CRT and hard copy (ref. 41). Extra care needs to be taken in the design of text displays so that the time to read text from a CRT is rapid and comprehension remains accurate.

GUIDELINES:

a. Text should be presented using upper and lower case characters.

RATIONALE: Reading time is faster with upper and lower case characters than with all upper case (ref. 75).

b. The default condition should be to left justify all lines of text, including the first word of each paragraph of text.

RATIONALE: Right justification with nonproportional spacing (fill justified) slows reading time (ref. 70).

c. Right–justification and fill–justification options should be available to the user.

d. Default text line spacing should be 150 percent of character height.

e. The default condition for line length should be between 52 and 80 characters.

RATIONALE: Line lengths of less than 52 characters result in slower reading times, but line lengths from 52 to 78 characters do not produce differences in reading time (ref. 20). However, 80 characters is more standard than 78 characters.

f. Users should have the ability to change the line length for an er.'ire text file or for any particular section of a file, down to a specific line.

g. The default values for the margins in a text file should be set to permit viewing of all of the characters in the entire horizontal line.

h. Any required dedicated function areas in a text file should be located in a consistent area.

i. Users should have the ability to set tabs for any particular section of a text file, including the entire file.

j. Users should have the ability to change the line spacing for an entire text file or for any particular section of a file.

4.1.1.2.2.2 DATA FORMS

DEFINITIONS AND DESCRIPTIONS: In data forms certain information is provided to the user and other information is entered by the user. DATA_FORMS are a user interaction tool which can support data entry and human–computer dialogue. For more guidelines on data forms as a human–computer dialogue technique, see subparagraph 4.2.8.1.3.

GUIDELINES:

a. Data forms should permit entry of predefined items into labeled fields of specially formatted displays. (ref. 64, p. 50.) If a user attempts to enter data into a data form in a manner that does not match the predefined format of the data form, the system should highlight the error and signal the user, e.g., with a beep.

b. Users should receive an error message only if they continue to make the same error. The error message should describe the proper manner for entering data. (See subparagraph 4.2.6.2 for additional information about error handling.)

c. Data fields should be defined by a label and character entry space (ref. 3, p. 91).

d. Labels should be left justified and end with a colon (ref. 64, sections 1.4.9, 1.4.16).

e. The label and the character entry area should be separated by at least one character space. At least two character spaces should be maintained horizontally between a character entry area and a succeeding label on the same row. For example,

Name: __________ Mail Code: _____.

f. Labels should be protected from inadvertent character entry (ref. 64, section 1.4.7; ref. 7, p. 92).

g. If appropriate, labels should cue the data entry, such as, "Date (mm/dd/yy): __/__/__" or "Cost ($): _____." (ref. 64, section 1.4.21.)

h. Provide cues to indicate the maximum length of a data entry field. For example, a broken underscore could be used to indicate the number of characters available for an entry (ref. 64, section 1.4.11).

i. The current field to be entered should be highlighted. (Key highlighting techniques include image reversal, i.e., reverse video. Note that underlining is reserved for indicating the length of a data field, so underlining should not be used for this purpose. For more information on highlighting, see subparagraph 4.1.1.3.1.)

RATIONALE: Irrelevant objects slow perceptual processing by competing for resources (ref. 23, ref. 33, ref. 69). Use of highlighting makes the current data field discriminable from irrelevant data.

j. When the data form is first opened, the placeholder cursor should be in the left–most position of the first available data field.

k. When users are highly familiar with a hard copy version of the data form, the data form should be formatted to be similar to hard copy source documents.

RATIONALE: Users should be able to transfer their previous training and experience with the hard copy format to the computer display.

l. Optional entries should be designated by TBD (ref. 64, section 1.4.12).

4.1.1.2.3 FORMAT OF TABULAR DISPLAYS

DEFINITIONS AND DESCRIPTIONS: Tables are especially useful for comparing the features of two or more alternative conditions. This subparagraph covers the format of two common types of tables — a matrix and a table containing functionally distinct areas (e.g., subtables). Although a user needs to read the information from both types of tables, they differ sufficiently in format that certain guidelines will apply to only one or the other. The matrix has a regular rows–by–columns structure in which the rows represent elements of a larger category and, similarly, the columns represent elements of another larger category. The data in the cells of the matrix are the values of the condition specified by the row element and the column element (see Figure 4–2 for an example). A spreadsheet and a correlation matrix are representative examples of a matrix. A user can obtain information from a matrix by scanning horizontally (rows across columns), vertically (columns across rows), and diagonally (rows and columns simultaneously). Figures 6–12, 6–15, and 6–16 contain examples of matrixes.

A table that consists of functional areas has a less regular structure than a matrix and, in many ways, may resemble a data form. However, unlike data forms, these tables can contain just data and not require any input. As a consequence of having less structure, such a table can present more varied types of information. The current Space Shuttle computer displays include tables made up of functional areas (see Figure 6–21). The user can obtain information from these tables primarily by reading across rows while moving down a column; the function of the column is principally to align and label the data.

4.1.1.2.3.1 STRUCTURE

4.1.1.2.3.1.1 ROW AND COLUMN

GUIDELINES: .

a. See the current release of NASA–STD–3000, Vol. IV, for current SSFP requirements on tabular data design.

b. The left–most column should contain the labels for the row variables (that is, the information by which the user will access other row items). (ref. 64, section 2.3.)

c. The top row should contain the labels for the column variables (that is, the information by which the user will access other columnar items). (ref. 64., section 2.3.)

d. Aids to increase row discrimination should be used in dense tables with many rows. For example, discrimination between rows can be accomplished by placing a blank line between a group of rows at regular intervals. (ref. 64, section 2.3.14.)

e. Spacing between rows should be consistent within a table and between related tables. (ref. 64, section 2.3.)

f. Locate items to be compared character by character with the characters to be compared aligned either vertically or horizontally. If comparing two rows of characters, one row should be above the other; if comparing two columns, one column should be immediate'y to the left of the other.

g. Each column should be discriminable from every other column. Discrimination between columns can be accomplished by having sufficient blank space (e.g., one or more character spaces) or a distinctive feature (e.g., a vertical line as a border) between columns. (ref. 64, section 2.3.)

h. Spacing between columns should be consistent within a table and between related tables. (ref. 64, section 2.3.)

i. The data within the rows and columns should be organized systematically (e.g., chronologically, alphabetically, by magnitude), when such organization is possible (ref. 64, section 2.3). For example, if a table for a life sciences payload displayed the growth of plants over a ten day period, the table might be structured as in Figure 4–2:

		DAY									
		1	2	3	4	5	6	7	8	9	10
PLANT Type	A										
	B										
	C										
	D										
	E										

FIGURE 4–2 EXAMPLE OF A TABLE

j. Alphabetic characters or numerical characters being used for a nominal (i.e., naming) function should be left justified within a column. Examples of a set of numerical characters used fcr a nominal function are Social Security numbers or equipment part numbers.

k. Numerical characters that represent a value on a numerical scale (i.e., a scale being used to represent a continuous range of numbers rather than a nominal function) should be aligned by their decimal points. When whole numbers are displayed, a decimal point should be assumed.

4.1.1.2.3.1.2 FUNCTIONAL GROUPS

GUIDELINES:

a. Related data should be displayed in groups which subtend five degrees of visual angle or less (ref. 72). For example, a related group might contain all items that can be activated during a reboost operation (task functional relation) or might contain all data entry items (table functional relation). Groups should be visually distinct from one another (e.g., by separating them from other groups with blank spaces)

b. For improved readability, overall screen density for tables should be no greater than 30 percent. Overall density is the ratio of screen character spaces filled to the total number of spaces. (ref. 72, ref. 19.)

4.1.1.2.3.2 ELEMENTS OF TABLES

4.1.1.2.3.2.1 CHARACTERS

General guidelines for characters are also presented in subparagraph 4.1.1.1.1.

4.1.1.2.3.2.1.1 ALPHABETIC CHARACTERS

GUIDELINES:

a. The sizes of alphabetic characters should be consistent within a table and between related tables.

b. The fonts and widths of alphabetical characters should be consistent within a table, except when a word or set of characters is highlighted by varying the typeface (e.g., through the use of italics or a "boldface" function).

4.1.1.2.3.2.1.2 NUMERIC CHARACTERS

GUIDELINES:

a. The display of numeric values should be applicable to the user's task, however, a user should be capable of selecting an alternative numbering system (e.g., scientific notation, log). The decimal number system should be the default display. (See the current release of NASA–STD–3000, Vol. IV, for current SSFP requirements on numeric display.)

b. A user should have the capability to select the precision with which a numeric value will be displayed, within reasonable system limitations.

c. Leading zeros in numeric entries for whole numbers should be suppressed (i.e., display 28 rather than 0028). A leading zero should be provided if the number is only a decimal, with no preceding integer (i.e., display 0.43 rather than .43).

d. The sizes of numeric characters should be consistent within a table and between related tables.

e. The fonts and widths of numeric characters should be consistent within a table. Highlighting (e.g., through the use of italics or a "boldface" function) the typeface should only be used when it does not change the column alignment within a table.

f. Information should be presented in a directly meaningful and immediately usable form. The user should not be required to mentally convert from one unit of measure to another. System aids for assisting the user in converting units should be considered.

4.1.1.2.3.2.1.3 ALPHANUMERIC CHARACTER GROUPS -

GUIDELINES:

a. When five or more alphanumeric characters without natural organization are displayed, the characters should be grouped in blocks of three to five characters, separated by a minimum of one blank space or other separating character such as a hyphen or slash. (See the current release of NASA–STD–3000, Vol. IV, for current SSFP requirements on alphanumeric grouping.)

4.1.1.2.3.2.2 LABELS AND HEADINGS

GUIDELINES:

a. Each group of data should be labeled (ref. 64, section 2.2). In a matrix table, the data groups are columns and rows. In a functional grouping table, the data groups are generally columnar functional groups.

b. Labels should be distinctive from the data (e.g., by underlining and writing in upper case).

c. A label should be separated from its associated data field by at least one standard character space, but should be close enough to the data field to allow the user to associate it with the appropriate data. (ref. 64, section 2.2.)

d. Labels should be separated from one another by at least two standard character spaces. (ref. 64, section 2.2.)

e. Labels should describe the data content of a data group accurately, using the fewest characters possible (ref. 64, section 2.2). Labels should consist of alphabetic and/or numeric characters, words, and/or abbreviations. A scheme for abbreviating labels should be followed consistently (see subparagraph 4.1.1.1.2.2).

f. The relation between a label and its referent should be consistent between tables.

g. The unit of measure (in standard abbreviated form) should be displayed as part of the column label. (ref. 64, section 2.3.)

4.1.1.2.4 FORMAT OF GRAPHICAL DISPLAYS

DEFINITIONS AND DESCRIPTIONS: A GRAPHICAL DISPLAY provides a pictorial representation of an object or a set of data. Graphical displays include line, solid object, and perspective drawings; bar, pie, and line charts and graphs; scatterplots; displayed meters; flowcharts and schematic diagrams; icons; and maps. The pictorial representation can be an analogue of the represented object or data (e.g., a picture) or can be symbolic (e.g., a schematic). In addition, certain graphical displays combine analogue and symbolic representations. For example, maps represent spatial information analogously to the area that is mapped, but can represent other data, such as city population, symbolically. Graphical displays also may include alphanumeric characters, for example, for labelling.

GUIDELINES:

a. Consider using a graphical display of data when users need to monitor changing data, quickly scan and/or compare sets of data. (ref. 64, sections 2.4.2 and 2.4.3.)

b. Categorical or trend data should be represented graphically.

c. Continuous data which can be categorized without a loss in information content should be represented graphically.

RATIONALE: Some studies have found that graphic displays seem to promote "holistic" processing whereas alphanumeric displays promote "serial" processing; this is especially true when there is time pressure (ref. 31, ref. 36, ref. 59). However, research findings have been equivocal and very task dependent. A large number of studies have found no significant differences in performance between alphanumeric and graphic displays (ref. 1, ref. 26, ref. 38). Tullis (ref. 72) found a speed advantage for graphic format which disappeared with extended practice. Powers, Lashley, Sanchez, and Shneiderman (ref. 55) found a speed advantage for tabular over graphic displays in a retrieval and comparison task. Therefore, judgment must be exercised in choosing a particular data format. Several formats should be developed and evaluated prior to choosing a specific format.

4.1.1.2.4.1 GLOBAL ORGANIZATION

GUIDELINES:

a. Graphical displays should maintain the visually simplest display consistent with their function. In general, the fewest lines or objects in a graphical display should be used. (ref. 71.)

b. Graphical displays should be designed so that a user notices the most important things first. For example, in constructing a graph, be aware that people notice heavier lines before lighter ones; brighter colors are detected before dim colors; and larger bars are detected before more slender bars. (ref. 39.)

c. The user should be able to view displays with more detail at request by a single action.

d. Graphical displays should optimize visibility, legibility, interpretability, and conspicuousness.

4.1.1.2.4.1.1 GRAPHS AND DATA CHARTS

GUIDELINES:

a. To the greatest extent possible, graphs and charts should display all of the relevant information and only the relevant information for the user to complete the current step in the task. The user should be able to request more detailed data with a single action. See Figure 4–3 for examples of graphs and charts.

b. The graphical display should direct the user's attention to the critical data (see also subparagraph 4.1.1.3).

c. The user should be able to enlarge (and subsequently to reduce) the graph, chart, or some subsection to "zoom in" on critical data.

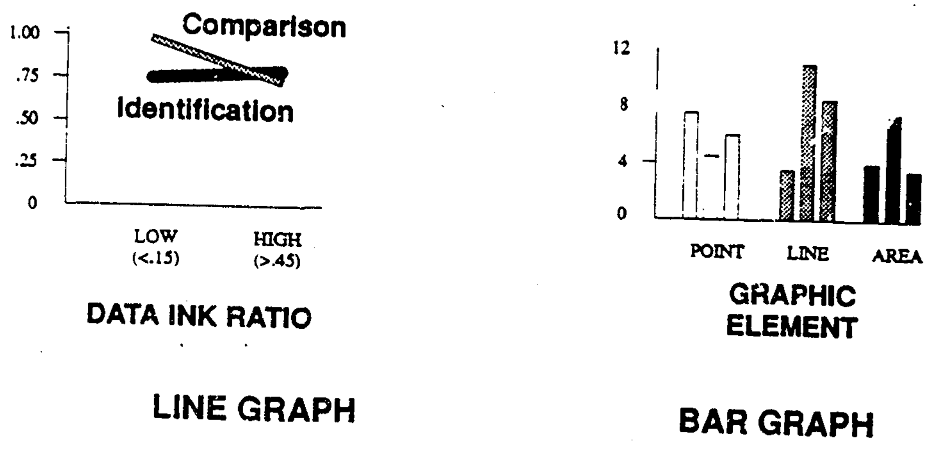
d. The user should be able to identify off–nominal values easily in tasks where there is a need to discriminate between such values. Coding techniques (e.g., flashing, reverse video) can be used to aid the user in discriminating between values. For more information on coding, see subparagraph 4.1.1.3.

e. To show how two variables are correlated or distributed, use a scatterplot (ref. 64, section 2.4.2).

f. Bar graphs should be used to show a comparative measure for discrete variables, for discrete levels within a variable, for a variable at different times, or to show apportionment of a total into its component parts (ref. 64, section 2.4.4).

g. Line graphs should be used to portray changes through time for one or more sets of data, such as trends over a period of hours, days, weeks, months, or years.

h. Column charts also portray data measured over time. Column charts can be more effective than line graphs in displaying a single set of data which covers a short period of time (e.g., data measure over a period of a week).



LINE GRAPH

BAR GRAPH

FIGURE 4–3 EXAMPLES OF DATA GRAPHS

4.1.1.2.4.1.1.1 THREE–DIMENSIONAL GRAPHS

TBD

4.1.1.2.4.1.2 DRAWINGS

TBD

4.1.1.2.4.1.3 METERS

TBD

4.1.1.2.4.1.4 DIAGRAMS AND FLOWCHARTS

GUIDELINES:

a. To the greatest extent possible, diagrams and flowcharts should display only the data required by the user. The user should be able to request more detailed data with a single action.

b. The user should be able to enlarge or reduce the size of a diagram or flowchart. The line scale should reflect the change in overall scale.

c. The display should direct the user's attention to the critical information on the flowchart or diagram (see also section 4.1.1.3, Highlighting).

d. Diagrams should show relative spatial relations accurately and should include a line scale describing the spatial relations.

e. A flowchart should be ordered according to its use. Generally, flowcharts are used to represent a temporal sequence of events; accordingly, a flowchart used to represent a temporal sequence should show the sequence spatially, with the first event on the leftmost side (or at the top) of the flowchart and subsequent events to the right (or below), in order (ref. 40).

4.1.1.2.4.1.5 MAPS

GUIDELINES:

a. Maps used for similar purposes should be displayed with a consistent orientation and reference points.

b. The default orientation for Earth maps should follow the normally accepted convention:

N

W E

S

c. The user should be able to select different orientations and reference points. The system should provide the user with a menu listing the common orientations and reference points.

d. Maps that display a large Earth area should project the Earth's curvature to minimize distortions (ref. 64, section 2.4.8.3).

e. Qualitative distributional maps should be used to display relative geographical locations of different kinds of data (e.g., government laboratories versus government centers across the United States) (ref. 56).

f. Quantitative distributional maps should be used to display variations in quantities across different geographical locations (e.g., to show the different sizes of populations of states across the United States) (ref. 56).

4.1.1.2.4.2 INFORMATION ELEMENTS AND LOCATION

GUIDELINES:

a. The label for a specific graphical object (e.g., a data graph, a schematic, or an icon) should be placed in close proximity to the graphical object (for example, directly below the object). The label locations for graphical objects should be consistent across all displays and the labels should not overlap.

b. Text labels of component parts of graphical objects (e.g., data points, curves, bars, symbols, map features) should be placed in close proximity to the part. When possible, the label should be on the component if it does not obscure the component. If multiple component parts of the graphical object are close to the label, a line should point from the label to the associated part.

c. Markings on a graphical display that have similar shapes, orientations, colors, and so on will tend to be grouped together, for example, in a legend associating a label with a content element such as a line or a shape (ref. 39).

d. Labels should be displayed in a normal orientation (ref. 64, section 2.4.11).

e. In addition to text labelling of objects or parts of objects, other techniques for coding the objects should be used, including coding by texture, color, or line type (see subparagraph 4.1.1.3). A legend should be provided for the translation of the code.

RATIONALE: Research shows that the use of a redundant physical code in addition to a text label may result in faster and more accurate performance with graphs than just the labeling (ref. 45).

4.1.1.2.4.2.1 GRAPHS AND DATA CHARTS

GUIDELINES:

a., Time or the postulated cause (the predictor or independent variable) should be plotted on the x–axis and the effect (criterion or dependent variable) plotted on the y–axis (ref. 64, section 1.6.1). The user should be provided with tools to help in creating and modifying graphs. For example, templates for typical axes, as well as typical scales for graphs, might be stored by the system and provided to the user with a single action.

b. When the data are far from the x– or y–axes, display an x–axis at the top and bottom of the graph and a y–axes at the left and right sides of the graph (ref. 64, section 2.4.1.8).

c. The use of broken axes in which the scale is discontinuous should be avoided.

d. Labels for both the x (abscissa) and y (ordinate) axes should both be displayed; the label for the x–axis should be below the axis, and the label for the y–axis should be to the left of the axis. (ref. 64, section 2.4.)

e. It is preferred that only one scale be used for an axis. However, this does not preclude the use of multiple scales for an axis when it is appropriate to the user's task and the information contained in the graph.

f. When two or more dependent variables are displayed in a line graph, the default condition should be that data be shown on separate two dimensional graphs. The graphs should be in the same display. However, when requested by the user, two independent variables may be displayed on a common graph as follows: the left y– and right y–axes would use different scales, as may the top and bottom x–axes. When different scales are used, the numerical values for the scales should be displayed in different type fonts.

g. At the user's request, the top and bottom x–axes may be displayed with different scales. When different scales are used, the numerical values for the scales should be displayed in different type fonts.

h. Numerical scales generally should have zero at the bottom as the first number on a vertical scale or at the left as the first number on a horizontal scale. The exceptions to this organization should be if the numbers are used for naming categories (ref. 64, section 2.4.1.6.), if zero is not a possible number on the scale, or if the scale contains negative numbers.

i. In constructing a scale, each scale should be marked off with ticks in equal units. The marking of the scale should begin from the point of origin, that is, where the x– and y–axes cross. For labeling tick mark divisions, a standard interval of one, two, five, or ten (or multiples thereof) should be used; between the tick mark divisions, intervening tick marks should be used to aid the user with visual interpolation. In special instance, the x–axis might be scaled in odd intervals to show customary divisions, such as the seven days in a week or the twelve months in a year. (ref. 64, section 2.4.1.5.)

j. Graphical symbols should be defined in a legend area and should be used consistently within a graph and between similar graphs. (ref. 64, section 2.4.)

k. Different variables or levels within a variable should be visually distinguishable. For example, on a bar chart, the different bars should have different colors or textures (see subparagraph 4.1.1.3.1.2 for guidelines on the use of color). Visually simple textures should be used to display differences (ref. 64, section 2.4.14).

l. Display of grid lines should be under the control of the user. The grid lines should be thinner than data curves and axes. Grid lines should not obscure data points, labels, or other objects in a graph.

4.1.1.2.4.2.2 DIAGRAMS AND FLOWCHARTS

GUIDELINES:

a. There should be a standard set of diagram and flowchart symbols. To the greatest extent possible, the symbols should be based on the standards for the task content. For example, computer programming symbols for decisions points, input, output, etc., should be available for software development tasks.

b. Flowcharts should use boxes to indicate events or states and lines to indicate relations between the boxes. The typical relation between events or states will be temporal. A state and an event may be displayed relative to a predetermined, displayed timeline (absolute time) or to another state or event (relative time) (i.e., Step 2 will be performed two minutes after Step 1).

c. Lines representing the temporal relation between events or states should have an arrowhead to show the temporal flow.

4.1.1.2.4.2.3 OTHER GRAPHICAL OBJECTS

GUIDELINES:

a. Drawings, meters, icons, and maps may be displayed at any appropriate screen location.

4.1.1.2.4.3 PARAMETERS OF GRAPHICAL DISPLAYS

4.1.1.2.4.3.1 GRAPHS AND DATA CHARTS

GUIDELINES:

a. Marks (e.g., lines) should be large enough to be noticeable and should be discriminable from one another, for example, by the use of different characters (e.g., ________, ___________, _ · _ · _). (ref. 39.) For more information on graphs, see subparagraph 4.1.1.3.3.1.2.

b. Adapt a consistent orientation (i.e., either vertical or horizontal) for bars displaying similar information in a series of graphs. (ref. 64, section 2.4.4.3.)

c. Lines should be clearly identified. The preferred method of identification should be placing the label in close proximity to the line (either within the graph or to the righthand side of the line). If a graph does not have sufficient room for labels, a legend should be used to display the line and its associated label.

d. A line graph should be formatted to be easy to read using the fewest number of lines as possible to convey the information.

e. Differences in color (e.g., red versus blue) should not be used to represent differences in quantities. (ref. 39.)

4.1.1.2.4.4 SCREEN AREA OF COVERAGE

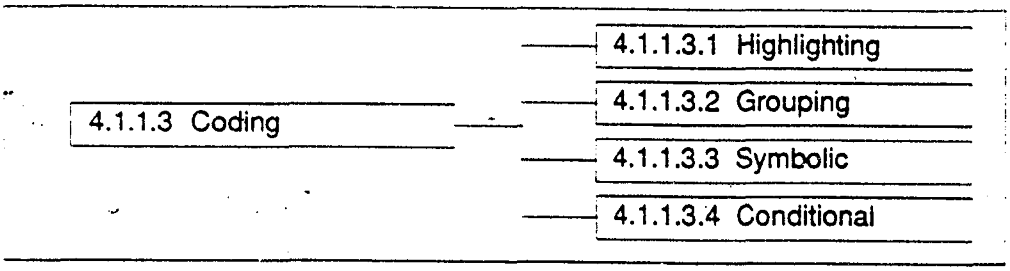
4.1.1.2.4.4.1 GRAPHS AND DATA CHARTS

GUIDELINES:

a. Graphs should exceed eight degrees of visual angle.

RATIONALE: Figures composed of distinct smaller elements are perceived holistically when less than eight degrees of visual angle in size. The perception of the smaller elements predominates when the figure is greater than eight degrees (ref. 51).

4.1.1.3 CODING



OVERVIEW OF SUBPARAGRAPH 4.1.1.3

DEFINITIONS AND DESCRIPTIONS: CODING is used for highlighting (i.e., to attract a user's attention to part of a display), as a perceptual indicator of a data group, or to symbolize a state or attribute of an object (e.g., to show a temperature level or for warning purposes). The guidelines for each of these three uses for coding are presented separately.

GUIDELINES:

a. A legend should be available to allow the user to interpret codes.

4.1.1.3.1 HIGHLIGHTING

DEFINITIONS AND DESCRIPTIONS: HIGHLIGHTING calls the user's attention to a feature of the display. Several highlighting methods are image reversal (reverse video), brightness/boldness contrast, color, underlining, blinking, flashing arrows, and changes in font. Examples of highlighting are shown in Figures 6–17 and 6–28.

GUIDELINES:

a. Highlighting should be restricted to only highly important information. For example, highlighting may be limited to the following:

— changed parameters of data critical to operation of Freedom Station subsystems, experiments or payloads;

— data that exceed accepted limits;

— any display item that indicates an abnormal condition;

— the position of a display item that must be provided or modified by the user before a process can continue; and .

— errors that would have a significant negative effect on the operation of any Freedom Station subsystem, as well as the associated user guidance to correct the error.

b. The purpose of highlighting may be only to direct the user's attention to a specific screen area, not necessarily to convey additional information.

c. Strive for consistency in using a particular highlighting method for related functions (e.g., reverse video for data entry errors).

d. All out–of–limit conditions should be coded consistently with the same highlighting method (e.g., boldness).

e. Highlighting of information should be minimized. A good rule of thumb for displays of nominal conditions is to limit the maximum amount of highlighting to ten percent of the display information. (ref. 3, p. 238.)

RATIONALE: If highlighting is to be used to attract the user's attention, the highlighting technique should be distinctive. If a large portion of a display is highlighted, the highlighting will no longer be distinctive.

f. When a display item is no longer important (e.g., after a critical error has been corrected), that item should no longer be highlighted.

4.1.1.3.1.1 IMAGE REVERSAL

GUIDELINES:

a. Reverse video should not be used when it has a detrimental impact on the user's perception of the display.

RATIONALE: Reverse video over large screen areas increases the perception of flicker. (ref. 14.)

b. Reverse video may be used to indicate selection of a parameter or set of data.

4.1.1.3.1.1.1 BRIGHTNESS/BOLDNESS CONTRAST

DEFINITIONS AND DESCRIPTIONS: Highlighting by brightness involves displaying information on a screen with light display features (e.g., characters or lines) on a dark background, with variation in the brightness of the light features. In contrast, boldness contrast involves dark display features on a white background, with variation in the darkness of the features.

GUIDELIᴵ:ES:

a. Highlighting with brightness should be limited to the following two levels: normal and overbright. Similarly, highlighting with boldness should be limited to the following two levels: normal and bold.

b. Brightness and boldness coding should be used to distinguish data entries in data forms from other display structures.

4.1.1.3.1.2 COLOR

GUIDELINES:

a. The use of color solely to direct the user's attention should be minimized.

b. If a particular color is used to direct the user's attention, that color should be used consistently for that purpose and should not be used for any other coding purpose within an application.

4.1.1.3.1.3 UNDERLINING

GUIDELINES:

a. Underlining should be used for highlighting text (e.g., titles, labels, and key terms in text).

b. Underlining should not be used when it has an impact on the perception of the display, for example, there should be space between the character and the underline mark.

4.1.1.3.1.4 BLINKING

GUIDELINES:

a. Blinking should be used solely to indicate a condition that requires immediate attention (e.g., extreme emergency or urgent action). The blinking cursor is an exception to this restriction.

b. Only a small area of the screen should blink at any time.

c. If an item must be read, blink a symbol marker next to the item, rather than the to–be–read item itself (see subparagraph 4.1.1.3.1.5).

d. Blink coding should not be used for displays requiring attention to detail or reading of text.

RATIONALE: Blink coding generally reduces search times, especially in dense displays. Equal benefit has been found if the entire stimulus is blinked, part of the stimulus is blinked or even if all of the nonstimuli are blinked. Visual search was not degraded, even in the nonstimulus blink condition (ref. 62 and ref. 63).

e. Blink coding should not be used for the entire display if detailed work is called for.

RATIONALE: Blink coding under these conditions results in a decrease in performance time due to deterioration in legibility.

f. The blink rate should be 2–5 Hz with a minimum duty–cycle of 50 percent (ref. 64, section 2.6.37).

g. The user should have the ability to suppress the blinking (e.g., after the user locates the message or data).

4.1.1.3.1.5 FLASHING ARROW

GUIDELINES:

a. A flashing arrow may be used to highlight a specific line of text or value in a table. The spatial relation between the arrow and the item to which the arrow points should be consistent (e.g., the arrow always to the left of the item). The arrow should not obscure any characters.

b. The rate of flashing should be 2–5 Hz with a minimum on–duration of 50 percent.

4.1.1.3.1.6 CHANGE IN FONT STYLE OR SIZE

GUIDELINES:

a. Highlighting of characters within a text file or a table may be accomplished by use of a different style of font (e.g., boldface, italics, or outlined characters in a plain text file) or the use of a larger size of font. Plain text should not be used as a highlighting method.

b. The use of a different font style should be preferred over the use of a different size for highlighting information.

c. The use of a different style or size of font should not obscure any nonhighlighted characters.

4.1.1.3.2 DATA GROUPING

DEFINITIONS AND DESCRIPTIONS: Grouping is a powerful technique for representing the similarity or commonality of data. Grouping can be accomplished by having similar data spatially close together in a display and/or by having similar data share a common perceptual attribute (e.g., color, shape, or size).

With perceptual grouping, search tends to be serial between groups and parallel within groups. Without perceptual grouping, a slower serial search of the entire display would be required (ref. 67 and ref. 68). In addition, perceptual grouping techniques reduce confusability among stimuli; like elements are seen as belonging to the same group, even though they may be spatially close to confusable stimuli (ref. 7, ref. 8, ref. 24, ref. 33, ref. 34).

GUIDELINES:

a. Grouping techniques (i.e., grouping by color, shape, spatial distance, orientation, type of character, etc.) should be used to group functionally similar information and to indicate membership in a common group.

b. Displays with high information density, should have an intermediate number of groups (i.e., 19 to 40 groups). If inherent functional groups of data exist, then they should be preserved.

RATIONALE: Performance with displays designed for search tasks is a nonmonotonic function of the number of data groups on the display. A display with a low number of groups lacks clear organization and is difficult to work with. A display with too many groups yields performance equally bad because the grouping information is dispersed. An intermediate number of groups yields the best performance (ref. 33, ref. 73).

c. Displays should provide cohesive groupings of display elements so that users perceive large screens as consisting of smaller identifiable pieces or chunks.

d. Spatial distance should be used for redundant coding when possible. Limitations are physical screen size and amount of information to be displayed.

RATIONALE: Although data grouping techniques have been found to be task dependent, grouping based on location has been generally successful across a variety of tasks.

e. Display items possessing two attributes [e.g., activated Thermal Control System (TCS) elements] should not be displayed between two groups of items which share one attribute each with the double–attribute item. For example, do not display a group of activated TCS elements between a group of activated Electrical Power System elements and a group of inactive TCS elements).

RATIONALE: Conjunctive stimuli which are displayed between two groups of elements sharing one dimension each with the conjunctive stimulus are very difficult to detect (ref. 67).

4.1.1.3.2.1 SHAPE

DEFINITIONS AND DESCRIPTIONS: Shapes (e.g., triangles, circles, squares) can be used to convey information about status (e.g., a triangle could indicate caution) or that elements of a data group are similar (e.g., circles would represent stars, hexagon would would represent meteors). See Figure 6–24 for examples of shape coding.

GUIDELINES:

a. Under most conditions, the preferred technique for data grouping in a graphic display should be shape coding, i.e., indicating similarity by having the same shape, provided the shapes used are large enough to permit discrimination of different shapes.

b. Shapes used in coding for data groups should be clearly discriminable. For example, the elements of one group in a display might be triangles and the elements of a second group might be circles.

c. To the greatest extent possible, assign shape codes based on established standards or conventions. (ref. 64, section 2.6.16.)

d. Consider limiting shape codes to 15 different shapes for a given display. (ref. 64, section 2.6.15.)

4.1.1.3.2.2 COLOR

GUIDELINES:

a. Color coding for group membership should be limited to five or fewer colors.

RATIONALE: While many studies have found an advantage for color displays in a search task (ref. 15), most studies agree that color is no better than any other form of coding (i.e. shape). The effect of color is primarily seen in increased motivation, attention, and task enjoyment (ref. 16, ref. 65, ref. 72).

b. Colors used for coding group membership should be highly discriminable (ref. 64, section 2.6.27). For example, on a light background, a choice of colors might include red, black, and green. However, on a dark background, the colors might include a desaturated red, green, and blue.

c. Color should be a redundant coding feature used to distinguish categorical differences between groups of data or functions.

RATIONALE: Color should be redundant with other stimulus features because failures in the color display hardware could put the monitor in a monochrome mode, ambient lighting conditions could interfere with the perception of color, and a relatively large percentage of potential system users may have deficits in color vision.

4.1.1.3.3 SYMBOLIC CODES

DEFINITIONS AND DESCRIPTIONS: A symbolic code is used not simply to attract the user's attention or to indicate similarity but to communicate the meaning of a display structure to the user. For example, the colors red and blue might be used to communicate heat and cold, respectively.

GUIDELINES:

a. Symbolic codes should be consistent with the requirements described in SSFP documents.

b. Display elements should be designed such that conflicting information is not communicated (i.e. make sure "LEFT" is not shown on the right, "UP" is not on the bottom, "LOW" is not on the top, "STOP" is not written on a green background, etc.).

c. If potentially conflicting information must be presented, spatial distance should be used to separate the conflicting elements, thereby decreasing the effect of the conflict.

4.1.1.3.3.1 GRAPHIC SYMBOLS

4.1.1.3.3.1.1 ICONS

DEFINITIONS AND DESCRIPTIONS: ICONS are pictorial, pictographic, or other graphic representations of objects or actions. Because the purpose of icons is to reproduce real–world items, icons are best suited to represent concrete nouns (ie., objects) or actions (ref. 58).

GUIDELINES:

The SSFP should have a standard set of icons for computer displays which are developed based on the following guidelines:

a. Each icon should represent a single object or action.

b. Icons should be perceptually simple graphical objects.

c. Each icon should be easily discriminable from all other icons and all display structures in use on the same display as that icon. The distinguishing feature between icons should be the external geometric configuration of the icon (ref. 45).

d. Whenever possible, icons should be pictographic (i.e., should be visually similar to the object or action that they symbolize), except when nonpictographic icons are the accepted convention (for example, engineering symbols for resistors or inductors).

e. When pictographic icons are not possible (e.g., when the object does not have a pictographic equivalent), nonpictographic icons should be designed so that functionally similar objects or actions are represented by perceptually similar icons.

f. The user should have access to a glossary that contains a list of the standard icons and their associated objects or actions through the on–line Help system.

4.1.1.3.3.1.2 DATA DISPLAY CODES

DEFINITIONS AND DESCRIPTIONS: A DATA DISPLAY CODE consists of graphical objects that represent data in a graph, diagram, or map. An example of a data display code is the use of different shapes of objects to plot data from different groups.

GUIDELINES:

a. Data display codes should be used only if two or more conditions need to be represented in a display that consists of a graph, diagram, or map.

b. Data display codes should be consistent within a display and between related displays.

c. Each data display code should be highly discriminable from all other objects in that display.

d. The data display codes should be labelled according to the guidelines in subparagraph 4.1.1.2.3.2.2

4.1.1.3.3.2 COLOR

GUIDELINES:

a. Choice of color for symbolic coding should be consistent with highly overlearned associations (i.e., red as a symbol for danger or stop, green as a symbol for go, etc.). The system should make the standard list of color meanings available to the user (e.g., in a legend on the display or through on–line Help).

b. Color should be a redundant coding feature.

c. The maximum number of colors to be used for coding on a given display should be five.

RATIONALE: When expanded beyond this, the coding value of color is lost and the display yields performance equal to that of a display with no color coding (i.e., a monochrome display) (ref. 13).

d. For user selectable colors, the user should not be able to select the same color for both foreground and background.

RATIONALE: The user might inadvertently select the foreground and background colors to be the same, thereby making the screen unreadable.

Table 4–1 provides performance data for both a search and an input task using a number of color combinations and by contrast reversal. The performance data is expressed in terms of percentage of error rate. The color combinations are listed by lowest (top) to highest (bottom) error rates. Overall, subjects' performance was better with the color combinations of black on blue, blue on a dark white, yellow on black, and a bright white on green. (ref. 53.)

TABLE 4–1 ERROR RATES FOR SPECIFIED COLOR COMBINATIONS

	Ordered Search Task Color Combinations		Ordered Address Task Color Combinations	
% of Error Rate Range	Positive Contrast Light/Dark	Negative Contrast Dark/Light	Positive Contrast Light/Dark	Negative Contrast Light/Dark
		Black/Blue	White2/Red	
		Red/Brown		Black/Blue
		Blue/White1		Blue/Brown
	Blue/Black			Red/White1
		Black/Brown		Blue/White1
25%	Yellow/Black		White2/Blue	
	White2/Black		Yellow/Blue	
		Magenta/Brown		Red/Brown
	Green/Magenta		Yellow/Black	
		Black/Green	Yellow/Red	
		Magenta/White1	Green/Black	
50%	White2/Magenta			Black/White
	Yellow/Red		White2/Black	
	Yellow/Blue		Blue/Black	
		Blue/Brown	White2/Magenta	
		Red/White1		Black/Brown
	White2/Red		White2/Green	
75%		Black/White1		Magenta/Brown
	Yellow/Magenta		Green/Magenta	
	White2/Blue			Green/White
	Green/Blue		Yellow/Magenta	
	White2/Black			Magenta/Green
		Green/White1		Black/Green
		Magenta/Green		Magenta/White1

4.1.1.3.3.2.1 COLOR BRIGHTNESS

GUIDELINES:

a. Changes in color brightness should be used to indicate changes in the intensity of the symbolized data parameter. For example, if temperature level is symbolized with red as above zero degrees and blue as below zero, increasing temperatures above zero should be indicated by increasing brightness of red and decreasing temperatures below zero should be indicated by increasing brightness of blue.

b. Changes in color brightness should be coded redundantly.

c. Differences in color brightness should be used only for relative judgments of parametric values (e.g., A is hotter than B), not for absolute identification of a parametric value (e.g., A is 125° F).

4.1.1.3.3.3 ALPHANUMERIC

DEFINITIONS AND DESCRIPTIONS: ALPHANUMERIC CODES are sets of letters and/or numbers used to identify a group of data (e.g., in a table or on a statistical graph). Acronyms and abbreviations can be considered to be one type of alphanumeric code. The guidelines for abbreviations and acronyms were listed in subparagraph 4.1.1.1.2.2 and will not be repeated here.

GUIDELINES:

a. Whenever possible, a meaningful alphanumeric code should be used in preference to a nonmeaningful code. For example, an acronym should be used instead of an arbitrary set of letters and numbers.

b. If an arbitrary alphanumeric code is used, the alphabetic and numeric components should be grouped within a code with a consistent length and format. Example: RL58 not 2M9Z.

c. Avoid the use of the letters O and I in an alphanumeric code, they are easily confused with the numbers 0 (zero) and 1 (one), respectively.

d. All alphanumeric coding should be displayed consistently in either upper or lower case but should not be mixed case.

e. Alphanumeric codes should have no punctuation except for codes which may be confused with words.

4.1.1.3.4 CONDITIONAL CUES

DEFINITIONS AND DESCRIPTIONS: CONDITIONAL CUES provide the user with information about the rules that operate under the current conditions. For example, the color red might be used to symbolize danger in a caution and warning condition and to indicate heat in a TCS display. Thus, the meaning associated with red is conditional. However, a system user would need information to indicate which condition, danger or heat, was in effect. This subparagraph describes guidelines designed to provide conditional information.

GUIDELINES:

a. If a symbol (i.e., a color, shape, word. acronym, abbreviation) has more than one meaning within the SSFP user interface, a conditional cue should be provided to disambiguate the meaning of that symbol.

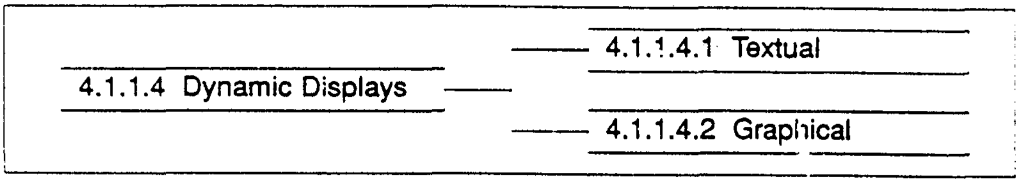
b. Within an application a symbol should never have more than one meaning.

c. One method by which the system should help the user to interpret the meaning of a symbol is by the use of display titles and labels that indicate the relevant condition. For example, a display in which red symbolized danger should be clearly titled as involving C&W. In contrast, a display in which red symbolized temperature should be titled as involving Thermal Control.

d. In addition to labels, displays should use legend(s) to indicate the meaning of all symbols within the display. The legend(s) should indicate the meaning of all symbols in a display, both symbols associated with multiple referents and those associated with only a single referent. The legend(s) should be clearly associated with its window and distinguishable from the display information.

e. More than one legend should be provided if a display has many different symbols. If multiple legends are used on a display, the symbols should be grouped in legends according to perceptual similarity (e.g., a color legend and a shape legend).

4.1.1.4 DYNAMIC DISPLAYS



4.1.1.4 Dynamic Displays

4.1.1.4.1 Textual

4.1.1.4.2 Graphical

OVERVIEW OF SUBPARAGRAPH 4.1.1.4

DEFINITIONS AND DESCRIPTIONS: DYNAMIC DISPLAYS contain display structures which change one or more feature(s), e.g., numerical value, color, shape, or spatial location, in real time or near real time. Examples of dynamic screen structures are tables that update data frequently, graphs or charts that update data frequently, and animated graphical objects. This subparagraph will not address general guidelines concerned with system response time. For those guidelines, see subparagraph 4.2.1.

4.1.1.4.1 TEXTUAL AND TABULAR DISPLAYS

GUIDELINES:

a. The system, rather than the user, should generally monitor all critical parameters and alert the user when the the values are out of range. In addition, the system, rather than the user, should monitor parameters that change very rapidly or very slowly.

RATIONALE: Users may not be able to perceive the values of changed parameters if the update rate is very fast. For slowly changing data, fixation on a display for an extended period of time (1–10 minutes) may result in selective adaptation. This lowers sensitivity to a stimulus similar to the one fixated upon. Long term fixation to an unchanging display may result in delayed reaction time to an updated stimulus which is similar in form (ref. 66).

b. The minimum update rate should be a function of the user's task. The minimum update rate should be determined through an analysis of the user's informational needs and rate of cognitive processing of the information by the user for each type of task (see also ref. 3).

c. The maximum update rate should be determined by the time required for the user to identify and process the changed feature of the display. The maximum update rate will also depend on the user's task.

d. A display "snapshot" should be provided to allow a user to examine any display selected at any time for a longer period. (ref. 3, p. 287.)

e. When the requirements of an operation–monitoring task dictate that current data changes be continuously viewed, the user should have the option of simultaneously viewing the "snapshot" display and the continuous display. (ref. 64, section 2.7.3.5.)

f. Update rates should be coordinated with any other tasks such that the user spends no more than two seconds waiting for a value to change.

4.1.1.4.2 GRAPHIC DISPLAYS

GUIDELINES:

a. Continuous changes should be used to present real–time data or to show trends in recorded data. Discrete changes should occur at the user's request or as some standard increment is reached by real–time data values.

b. Items on a graphic display should not move faster than 60 degrees per second of visual angle with 20 degrees per second preferred (ref. 3, p. 286).

RATIONALE: During motion, gross visual attributes and spatial orientation are usually preserved while small details may be lost or processing slowed. Perception of fast moving stimuli may be incomplete (ref. 66).

4.1.1.5 CAUTION AND WARNING DISPLAYS

GUIDELINES:

See the current release of NASA–STD–3000, Vol. IV, and NASA–STD–3000 for current SSFP requirements on classification of C&W messages.

a. C&W information should be presented through both the visual display and an auditory display.

RATIONALE: Redundant information is most likely to be perceived by the user, no matter the task that he or she is doing. Because users may need to refer to emergency information over a time period of several seconds to several minutes, the information should be displayed visually. In addition, due to the nature of emergencies, a visual medium is likely to be very effective for presenting information concerning an emergency (ref. 80). The positive aspects of speech presentation are the omnidirectionality provided by auditory information and the temporal organization inherent in speech which provides for additional contextual information. With this contextual information, users can identify specific information easily.

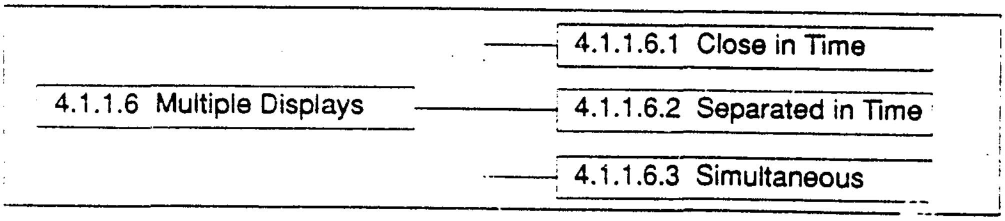
b. The visual display of emergency information should be redundant, using pictures, schematics, color, and text.

c. Pictures used for emergencies should be preceded by an auditory alerting tone.

d. Pictures used for emergencies should consist of a series of presentations which present temporal/hierarchical organization information (e.g., problem system, problem subsystem, and description of emergency state) (ref. 57). However, the user should have the ability to select the information (system, subsystem, or state) that he or she needs in the emergency.

RATIONALE: The "zooming in" feature provided by the "system to problem" order lends more information to the user since each successive presentation is a subset and therefore somewhat redundant with the previous presentation. This serves to re–emphasize the context and i, an important feature of pictures that is not present in speech (ref. 57).

4.1.1.6 AIDING THE USER IN INTERACTING WITH MULTIPLE DISPLAYS



4.1.1.6 Multiple Displays

4.1.1.6.1 Close in Time

4.1.1.6.2 Separated in Time

4.1.1.6.3 Simultaneous

OVERVIEW OF SUBPARAGRAPH 4.1.1.6

DEFINITIONS AND DESCRIPTIONS: Users frequently must work with several different displays simultaneously or in close succession. For example, during the operation of a Life Sciences payload, a user might need to examine instrument health status, the human or animal subject status, and the data being collected in rapid succession. In other cases, the user might have to interact with several very similar displays. In addition, users may also be involved in several different tasks at once (e.g., in a multitasking environment). As a consequence, a long time may intervene between looking at successive displays for a given task. The HCI should aid the user in integrating information across displays when required but also should permit the user to maintain a separation between information from various displays. These guidelines could only be implemented for a set of displays that were designed to be used together as a group; such a design of multiple displays should only be derived from a detailed analysis of the user's task.

4.1.1.6.1 MULTIPLE DISPLAYS CLOSE TOGETHER IN TIME

GUIDELINES:

a. A sequence of displays with which the user will have to interact in close temporal proximity should be contained in separate windows which can be displayed simultaneously or nearly simultaneously (see subparagraph 4.2.3.3 for window management guidelines).

b. When the user is working with a designed sequence of displays to perform a well–defined temporal sequence of task steps, the user should be provided with a position reference for his or her location within the display sequence.

4.1.1.6.2 FUNCTIONALLY–RELATED DISPLAYS SEPARATED IN TIME

GUIDELINES:

a. The capability to accept and maintain information, independent of application, should be provided for holding relevant information across displays or applications. An example of this capability can be seen in subparagraph 4.2.4.4.2 on the excerpt file.

b. Management of temporally discontinuous displays should be facilitated through the use of windows (see subparagraphs 4.1.1.1.5 and 4.2.3).

c. Information about a user's position in a functional hierarchy of tasks or task steps and their related displays should be provided, for example, by graphical representations.

d. Users should be able to move to any other display in the display sequence (e.g., by selecting a graphical representation of the display for a given task step). When actions on one display in a sequence require completion of actions on a previous display, the user should be able to move to a display only when all of the conditions have been met or an intentional override procedure has been completed.

4.1.1.6.3 INFORMATION PRESENTATION TO THE USER DURING SIMULTANEOUS TASK PERFORMANCE

GUIDELINES:

a. Whenever several tasks must be performed simultaneously (multitasking), and the tasks are independent (i.e., do not require integration), the tasks should be designed to present information to different sensory modalities (i.e., auditory versus visual) or to use different types of cognitive processing resources (i.e., verbal versus spatial). For example, a visual tracking task and search for a particular icon (independent tasks) should not be concurrent because both are spatial in nature. In contrast, verbal communication with the ground and search for a particular icon could be performed simultaneously, if necessary.

b. Whenever multitasking involves tasks that are not independent (i.e., require integration), the tasks should be designed to access the same pools of processing resources (i.e., verbal, spatial, auditory, visual, etc.). For example, if two values must be compared (integration), both values should be displayed visually, rather than one displayed visually and the other in the auditory dimension.

RATIONALE: Time–sharing is less efficient if the two tasks require resources from the same pool (ref. 78, ref. 79). When the tasks require integration of information, performance is benefitted if the same pool of resources is accessed (ref. 80).

4.1.1.7 DISPLAY BACKGROUNDS

GUIDELINES:

a. Because the background color on a display does not present any information, it should not distract the user from the data.

b. The background color should be of an appropriate hue/contrast to allow the data (foreground) to be easily visible.

c. Only one background color should be used on a display.

RATIONALE: Background color can influence the way a user interprets a color symbol (e.g., shape, lines). When a color symbol or figure is surrounded by another color, the surrounding color can change the appearance of the enclosed color. For example, green on a yellow background will appear more blue than the same shade of green on a blue background.

4.1.2 HARD COPY PRESENTATION OF INFORMATION

GUIDELINES:

a. Printer delay shall be no more than 1 to 2 seconds to acknowledge a command if the user is interfacing with the computer through the printer. (See the current release of NASA–STD–3000, Vol. IV, for current SSFP requirements on printer response.)

b. A print malfunction message shall be provided to alert the user when printing that has been requested is not being done due to a malfunction. (See the current release of NASA–STD–3000, Vol. IV, for current SSFP requirements on printer malfunction messages.)

c. Users should have the ability to print the information on a display exactly as it appears on the screen with a single user action.

d. Users should have the ability to print the contents of a selected window.

e. Users should have the ability to print an entire data file with a single action. The user should be able to select the file to be printed when he or she is working in the file and when he or she is outside of the file.

f. Users should have the ability to save the contents (e.g., layout, graphics, data) of an entire display which can be printed at a later time. Saving the display's contents will require only a single user action.

g. Users should have the ability to print selected pages of a data file.

h. Users should have the ability to perform other computer interactions/tasks while the system is printing. If the user makes changes on a file as that file is being printed, the printed copy should not reflect those new changes.

i. Users should have the ability to reformat files for printing in a limited manner. Examples of reformatting functions include changing the orientation of the hard copy from vertical to horizontal or vice versa, changing the font size or type of the file, changing the spacing between lines, and inserting blank lines or lines containing characters at the top and/or bottom of each page.

j. Users should be able to get a printed log of all of his or her interactions with system performed during a designated period or task.

k. Users should be able to direct the system to digitize hard copy data with a single action. Digitized data may be saved, reprinted, or displayed.

4.1.3 AUDITORY PRESENTATION OF INFORMATION

DEFINITIONS AND DESCRIPTIONS: AUDITORY PRESENTATION OF INFORMATION includes both verbal information presented by speech (either recorded or electronically created) and nonspeech sounds, e.g., bells, whistles, beeps. Auditory presentation of information has several limitations:

— Information presented by speech involves a temporal string; consequently, a user's memory may be taxed if the verbal string takes a long time due to either the amount of information or long interword intervals.

— Sounds are frequently difficult to localize, which would create problems if localization of information is important.

4.1.3.1 SPEECH

GUIDELINES:

a. Speech displays of information should be considered only when the following conditions are met: the environment has low ambient noise, the user's attention is not directed toward a visual display, and the use of a visual display is impractical. (ref. 64, section 4.0.26.)

b. Speech display should be used to provide only a limited number of messages, the messages should be short and simple. (ref 64, section 4.0.29.)

c. When information is presented by a speech display, the user should be able to replay the speech message as many times as needed.

d. When speech display is used to provide warnings as well as other forms of information, ensure that spoken warnings are easily distinguishable from routine speech displays. For example, add a special alerting signal to introduce the auditory warning display. (Ref 64, section 2.6.42.)

e. Consider presenting both visual and speech displays redundantly. Speech provides for the display of information without distracting the user from their primary task whereas, visual displays provide the user with the capability to refer back to the information thereby, improving accuracy of information transfer. (ref. 81.)

4.1.3.2 NONSPEECH AUDITORY SIGNALS

GUIDELINES:

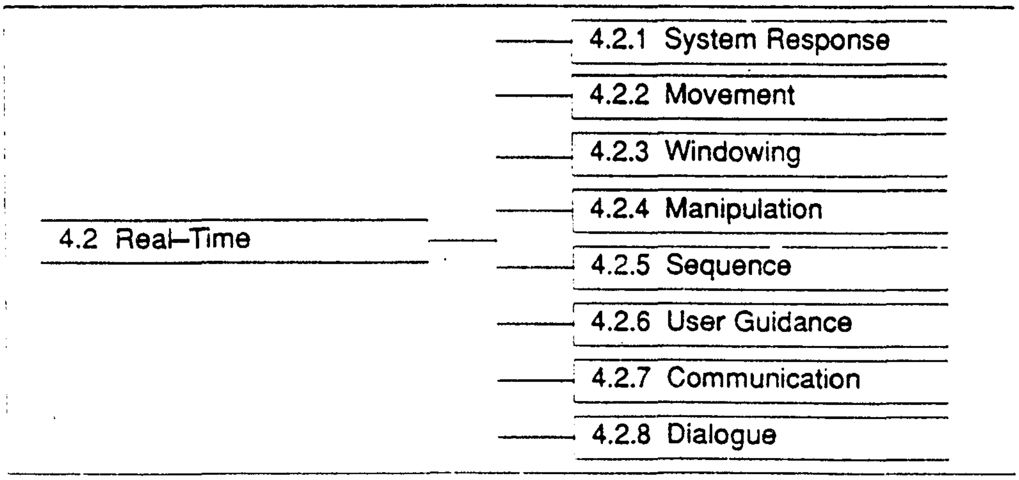
a. Auditory alerts, as well as C&W sounds, should be redundant with visual warnings presented by pictures, messages, and blink coding (see also subparagraph 4.1.1.3).

b. Users should have the capability to selectively disable noncritical auditory signals.

4.1.4 FORCE REFLECTIVE PRESENTATION OF INFORMATION

TBD

4.2 REAL–TIME INTERACTIONS BETWEEN USERS AND THE SPACE STATION FREEDOM PROGRAM COMPUTER SYSTEMS



OVERVIEW OF PARAGRAPH 4.2

GUIDELINES:

The basic philosophy of this paragraph can be summarized in a single guideline: To the greatest extent possible, the SSFP computer systems should be designed so that, when a user interacts with the SSFP computer systems, the outcome of the interaction should be what the user intended. This guideline manifests itself throughout the user interface. One concrete example is from a "print" transaction, in which the user commands the system to print a hard copy representation of data from the system (e.g., a data file). The copy that the system prints should be identical to the copy that the user sees prior to printing. This example might be implemented in one of several different ways. For example, the system might provide the user with a limited ability to preview what he or she will get in hard copy. In contrast, the system might only display to the user exactly what he or she would get in a hard copy format. Other relevant examples can be found throughout this paragraph.

4.2.1 SYSTEM RESPONSE TIME

GUIDELINES:

a. System response time requirements are in Table 4–2. (ref. 3 and see the current release of NASA–STD–3000, Vol. IV, for current SSFP requirements on system response time.)

b. The SSFP computer systems should respond to user commands and requests in minimal time.

c. Response time for a new display involving access of different files or transformation of data may take two to ten seconds. (This is acceptable because the user will likely understand that the commanded operation is complex.) (ref. 64, section 2.7.1.)

d. The variability of response times should be kept to a minimum. Response time deviations should not exceed more than half the mean response time (e.g., if the mean response time is four seconds, the variation should be limited to a range of three to five seconds). (ref. 3.)

e. A message should be displayed if response times are long and there is no indication to the user that the system is processing. Additionally, as the status of the system changes during a multiple–step task, the status message should change accordingly.

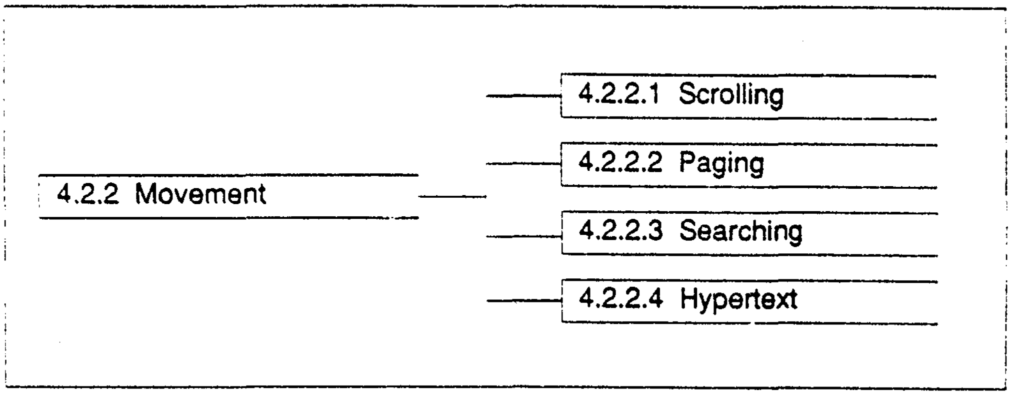
TABLE 4–2 PREFERRED AND MAXIMUM SYSTEM RESPONSE TIMES
(PAGE 1 OF 2)

User Activity	Maximum Response Time (Sec.)	Preferred Response Time (Sec.)
Control Activation (for example, keyboard entry, cursor controller movement)	0.10	<=0.10
System activation (system initialization)	3.0	<=0.50
Request for given service:		
Simple	2.0	<=0.25
Complex	5.0	<=2.0
Loading and restart	15.0 – 60.0	<=6.0
Error feedback (following completion of input)	2.0	<=0.25
Response to ID	2.0	<=0.25
Information on next procedure	<5.0	<=2.0
Response to simple inquiry from list	2.0	<=0.25
Response to simple status inquiry	2.0	<=0.25
Response to complex inquiry in table form	2.0 – 4.0	<=0.25
Request for next page	0.5 – 1.0	<=0.25
Response to "execute problem"	<15.0	<=6.0
Light pen entries	1.0	<=0.25
Drawings with light pens	0.1	<=0.10
Response to complex inquiry in graphic form	2.0 – 10.0	<=0.25

TABLE 4–2 PREFERRED AND MAXIMUM SYSTEM RESPONSE TIMES (PAGE 2 OF 2)

User Activity	Maximum Response Time (Sec.)	Preferred Response Time (Sec.)
Response to dynamic modeling	—	—
Response to graphic manipulation	2.0	<=0.25
Response to user intervention in automatic process	4.0	<=1.50

4.2.2 MOVEMENT THROUGH DISPLAYS



4.2.2 Movement

4.2.2.1 Scrolling

4.2.2.2 Paging

4.2.2.3 Searching

4.2.2.4 Hypertext

OVERVIEW OF SUBPARAGRAPH 4.2.2

DEFINITIONS AND DESCRIPTIONS: One of the most frequent interactions between a user and a computer is movement from one display to another. Frequently, this movement through displays involves a sequence of linked displays that show information from a single object or file. For example, a user may need to move from one page of text to another in a word processing or text editing task, from one form to another in a data form filing task, or from one graphic display to another when either producing or examining graphics.

Scrolling and paging are two of the most common methods for display movement. SCROLLING involves the continuous vertical or horizontal movement within a set of linked displays (e.g., a text file). In contrast, PAGING involves the discrete movement within a set of linked displays. The unit of movement in paging generally is one display. Note that scrolling and paging cannot be done simultaneously. Guidelines describing scrolling and paging display structures are given in subparagraph 4.1.1.1.6.5.

GUIDELINES:

a. Users should have the ability to move through a set of linked displays by both scrolling and paging, although users should be able to use only one method for movement at a time.

4.2.2.1 SCROLLING

DEFINITIONS AND DESCRIPTIONS: Humans perceive two types of scrolling: MOVING TEXT, in which the information in the display (e.g., text) appears to move over a fixed display window, and (2) PANNING, in which a window appears to move over a fixed display of information.

GUIDELINES:

a. Only one method of scrolling (moving text or panning) should be implemented, with panning being the preferred method (ref. 64, section 2.7.2; ref. 3, p. 165).

b. Users should have access to several different means by which they may scroll (e.g., a scroll bar, keyboard arrow keys, and keystroke commands).

c. The scroll motion rate should allow the user to scroll by line or by display unit. Either technique should provide a smooth flow of text.

d. The direction that a user will be scrolling (toward the top or bottom, left or right) should be evident to the user before he or she begins the scroll sequence. For example, scroll arrows on a scroll bar might point in the direction that corresponds to the scroll direction.

4.2.2.2 PAGING

GUIDELINES:

a. Users should have the ability to page using several different techniques (e.g., paging by means of moving a page icon on the scroll bar or by the use of a dedicated function key for paging forward and a dedicated function key for paging back through a file).

b. The system should provide the user with information about his or her position in a sequence of linked displays. For example, a page icon on the scroll bar might display the page number of the current page.

c. Users should be able to page in one page or multiple page increments. For example the user might page multiple pages directly by moving the page icon on the scroll bar at which time the display might move to the location in the file that corresponds to the page number on the page icon.

d. The direction that a user will be paging (toward the top or bottom, left or right) should be evident to the user before he or she begins to page. For example, scroll arrows on a scroll might point in the direction that corresponds to the paging direction.

e. The movement of the file should be discrete with no display of intermediate pages between the starting page and the selected page.

4.2.2.3 SEARCHING

GUIDELINES:

a. Users should have the ability to search for and move to a specific line number in a file.

b. Users should have the ability to search for and move to a literal string of alphanumeric characters.

c. Users should have the ability to find multiple occurrences of a literal string of alphanumeric characters.

d. Users should have multiple methods by which they can engage in searching for lines or alphanumeric strings (e.g., by means of either a menu or a UIL command).

4.2.2.4 HYPERTEXT

DEFINITIONS AND DESCRIPTIONS: HYPERTEXT is a data retrieval, data input, and data management structure in which nodes of data (in the form of text or graphics) are joined by links. As a method of gaining non–linear access to electronic information, Hypertext has a function similar to methods of non–linear access to hard–copy data, such as tables of contents, indices, and footnotes. However, with an appropriate user interface, Hypertext's capabilities can far exceed those of the hard copy–based methods.

Hypertext nodes and links may be accessed through either browsing or authoring. In browsing systems users may search through a database to obtain information contained in the nodes by following pre–established links between nodes. Information typically cannot be entered into the database in the browse mode, but can be copied to an external file. Authoring systems provide more flexible data manipulation capabilities than browsing systems. In an authoring mode, users may create, modify, or eliminate nodes and links, as well as add or delete information. Browsing systems are useful for data searches and information retrieval whereas authoring systems are often used as thinking, annotation, and writing tools.

Hypertext presents a number of potential benefits and pitfalls. Because Hypertext technology is relatively new, the user interface issues have not been fully documented. This subparagraph presents the advantages and disadvantages of Hypertext systems apparent at present and some guidelines regarding user interactions with them.

Advantages:

— Information may be organized into linear, non–linear, and hierarchical structures.

— Regardless of the organization of the data, it can be modified easily. Paths through links and nodes can be recorded, allowing developers to increase or eliminate links or restructure a hierarchy on the basis of link traversals and the frequency with which nodes are accessed.

— Flexibility — the same node can serve multiple purposes, increasing document customization and reducing redundancy.

— Powerful capability to rapidly check references, related items, and additional information.

— Significant advantages over paper in terms of weight and ease of updating.

Disadvantages:

— Difficult for users to maintain their sense of context — they may forget their place in the hierarchy and their available options.

— Reading from an electronic display is generally slower and/or less accurate than reading from paper (ref. 28).

— Hypertext requires more decisions on the part of the reader to determine which nodes should be read.

— Fixed browse organization — the user of a browsing system must follow the organization imposed by the developer, not necessarily the organization the user prefers.

— Difficult to annotate Hypertext documents.

— Hypertext relies on the untested basic assumption that the non–linear organization of data facilitates thinking and reading.

Hypertext browsing tools may be most appropriate for procedures that are complex and likely to be nonlinear; that is, those that frequently go into fault management mode or that have many conditional steps. Given well–constructed links, Hypertext also may be appropriate for searching databases and gaining quick access to information. Authoring tools appear to be more appropriate for restructuring or organizing information, for tasks that aren't well structured, and for tasks that can be divided into relatively small components. Currently, Hypertext tools probably aren't appropriate for editing tasks or tasks which are performed more poorly under interruption. Authoring tools aren't necessary for tasks that are strictly linear since many of the advantages of Hypertext are lost in this context. Thorough task analyses will help determine the functionality necessary for an application.

GUIDELINES:

a. Users should not have the power of authoring tools if only a browsing tool is needed.

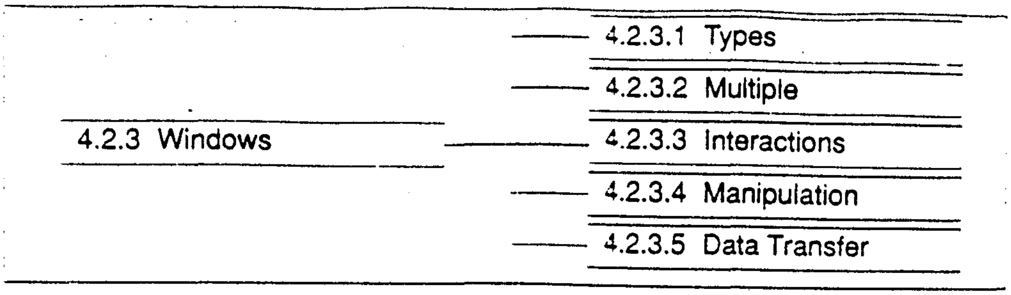
RATIONALE: Users may inadvertently change or destroy links. This possibility may be averted by building into critical files a "browse only" capability.

b. Items of information which are linked to other items or nodes should be coded distinctively and unambiguously.

c. Hypertext tools should always have a context–sensitive help function, including an overview function that displays the entire hierarchy and a history function which tells the user which paths have been travelled.

d. Hypertext browsing tools should generally act in a question and answer dialogue (ref. 30).

4.2.3 INTERACTIONS WITH WINDOWS



OVERVIEW OF SUBPARAGRAPH 4.2.3

4.2.3.1 TYPES OF WINDOWS

DEFINITIONS AND DESCRIPTIONS: WINDOWS may be categorized into two types: whether the window is open or closed and active or inactive. Closed windows require some action by the user in order to gain perceptual and functional access to the window. For example, a user may select and open an icon that represents a window or, in contrast, the user might input a UIL command to open a specific window. Open windows are both perceptually and functionally available to the user.

Two types of open windows exist: active and inactive. Active windows are defined functionally from the user's viewpoint: Commands issued by the user are directed to an active window. Typically, this means that an active window is currently receiving input from the user, has last received input from the user, or has been readied for input through the user's explicit action. In any event, the user is generally said to be "working in" the active window (processing a document, controlling a system, entering data, etc.).

Inactive windows are perceptually and functionally available to the user (the user can see and obtain information from them) but are not immediately available in the sense that the user must activate an inactive window before working in it.

This classification scheme best fits non–multitasking environments in which many windows may be open but only one is active. Multitasking environments allow the user to oversee tasks controlled from numerous windows simultaneously. These conditions permit multiple active windows and present the added difficulty to the user of routing commands to the appropriate active windows. One possible solution is to further classify active windows into interactive windows (active windows in which the user is currently working) and noninteractive windows (active windows in which processes have been initiated but are not the current focus of the user). Myers (ref. 50) has proposed a similar classification for windows under conditions of multitasking, calling the interactive window "the listener" or "the input focus."

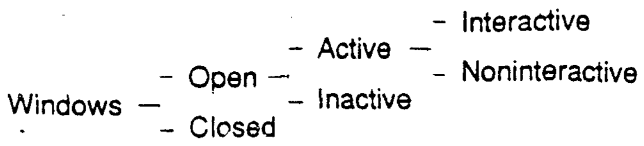


FIGURE 4–4 A CATEGORIZATION SCHEME FOR WINDOWS

A variety of methods for maintaining a distinction between multiple active windows (and the inactive or closed windows associated with them) exist. The system might maintain a directory of windows, a schematic illustrating the relationship between windows, or a coding or highlighting system to identify windows in a hierarchy.

The 'ICI issues regarding multitasking environments remain largely unexplored. The terms in the hierarchy described above (and presented graphically below) will be used to describe window states throughout this document. This simple classification is only a beginning. Developers should remain aware of advances in multitasking technology.

GUIDELINES:

a. Commands issued by the user should directly affect the interactive window.

b. Actions by the user should primarily affect the interactive window. However, actions in the interactive window may affect any other window.

RATIONALE: the user might want to use an inactive window as a display (or a closed window as the recipient) for the consequences of actions performed in the interactive window.

c. Window types should be perceptually distinct.

RATIONALE: Permitting changes in noninteractive and interactive windows must be done with great caution so that the changes are visible to the user in a layered window environment. For example, windows whose contents are changed by an action in another window might be brought to the front of the display, possibly as tiled windows with the active window.

d. Not all windows need necessarily have the full set of interactive capabilities. The capabilities present in a window should be a function of how the user will interact with the window. For example, a window that simply presents a one–line status message from the system that the user will only read and not respond to might need to only have the ability to be closed. It might not need to be resizable, movable, etc.

4.2.3.2 MULTIPLE WINDOWS

DEFINITIONS AND DESCRIPTIONS: Techniques to manipulate multiple windows include "tiling" in which multiple windows on the same display abut, but do not overlap and "layering" in which multiple windows overlap and obscure the contents of the covered windows. As the number of windows increases in the tiled window environment, the sizes of the windows generally decrease.

GUIDELINES:

a. When multiple windows are open simultaneously, the default condition should be a tiled window environment, provided that the size for each window is sufficient to hold usable, readable information (ref. 11). A suggested maximum number of tiled windows is four. The size of the tiled windows might be approximately 1/n of the available display, where n is the number of windows.

RATIONALE: Bly and Rosenberg (ref. 11) compared people's interactions with tiled and layered windows. They found that users spend a substantial amount of time performing window maintenance tasks (e.g., locating and "popping" a window to the front, closing windows, and moving windows) in a layered environment. However, the utility of a windowing environment is likely to be conditional on the user's task.

b. Users should have the capability to select between "tiling' and "overlapping" window environments.

c. Users should have the capability to obtain information about any and all open windows. At a minimum, this information should include window name, type, and any process initiated through and displayed in that window.

d. In a layered window environment users should have the capability of moving a window to the front of or behind any or all other windows.

e. When a tiled window environment results in windows of a size that would reduce the user's ability to use the information in the window, a layered window environment should be employed. The layered windows can overlap and should be the default window size until resized by the user.

f. Users should have the abiliry to move tiled windows so that they overlap.

4.2.3.3 INTERACTIONS WITH WINDOWS

4.2.3.3.1 INTERACTIVE WINDOWS

GUIDELINES:

a. The user should be able to put a window in the interactive state by performing any of a set of simple actions in that window or related to that window (for example, by moving the pointing cursor to the window and performing any action, including pressing a key or a button on a cursor control device, a command to open a specific window, selecting a window title from a list on a menu, selecting an icon for the window).

b. The action that puts a window into the interactive state should automatically place the placeholding cursor in that window so that the user can provide inputs through that window.

c. Under nominal operating conditions, interactive windows should have display priority over all windows (be frontmost on the display).

d. C&W windows should have display priority under emergency conditions.

e. Interactive windows in both the tiled and layered window environments should be perceptually distinct from noninteractive windows.

4.2.3.3.2 ACTIVE WINDOWS

GUIDELINES:

a. Active windows in both the tiled and layered window environments should be perceptually distinict from all other window types.

b. Active windows should have display priority over all but the interactive window.

c. To help the user to manage active windows, the system should provide user aids when multiple windows are active. An example user aid might be a list of active windows, including the name (title) of the window, the ongoing activity, the time elapsed since the beginning of the activity, the screen location of each window, the relation between activities in the various windows (e.g., if they are hierarchically related as part of the same task), and a menu that would allow users to view the contents of a selected window in the layered environment.

4.2.3.4 MANIPULATING WINDOWS

4.2.3.4.1 OPENING WINDOWS

GUIDELINES:

a. The user should be able to open a window by performing any of a set of simple actions (for example, by issuing a command to open a specific window, selecting a window title from a list on a menu, selecting an icon for the window).

b. The action that opens a window should automatically make that window active.

4.2.3.4.2 MOVING WINDOWS

GUIDELINES:

a. Window movement capability should be provided such that the user can move windows to different areas of the display.

b. Windows should not be movable to obscure menu bars, access to the command area, or C&W messages.

c. Notification should be provided to the user if a function area covered by a window requires immediate attention.

d. Users should not be allowed to move windows entirely off the display.

e. Windows partially moved off the display should be made readily accessible with a single action.

f. Movement of a window should appear to be smooth and continuous to the user.

4.2.3.4.3 CLOSING WINDOWS

GUIDELINES:

a. Users should be able to close a window with a single action.

4.2.3.4.4 SCROLLING WITHIN A WINDOW

a. Scrolling within windows should correspond to the guidelines in subparagraph 4.2.2.1.

4.2.3.4.5 RESIZING WINDOWS

DEFINITIONS AND DESCRIPTIONS: When a user resizes a window, he or she adjusts the external dimensions of the window but does not change the basic features of the window. Note that closing a window not only reduces the size of the window but also changes its features (e.g., a user cannot input into a closed window).

GUIDELINES:

a. Users should be able to change the horizontal and vertical dimensions of a window independently and with minimal effort.

b. Users should be able to select between two resizing conditions: resizing which does not change the size of the window contents, and resizing in which the size of the window contents increase or decrease with the changes in the size of the window. The default condition should be resizing which does not change the size of the contents.

RATIONALE: Many uses for resizing involve sizing the window to fit the graphical or textual contents of the window (e.g., when trying to fit a complex graphical object into a large enough window to permit good resolution of elaborate details). In such cases, the user would want the object to remain the same size while the window changed. In contrast, a user might want to tile a window (e.g., by decreasing its size and moving it to a corner of the screen), but still be able to see the contents of the window. In such a case, the user would want the window contents to change in size with the window.

c. The upper limit for resizing a window should be the size of the computer screen.

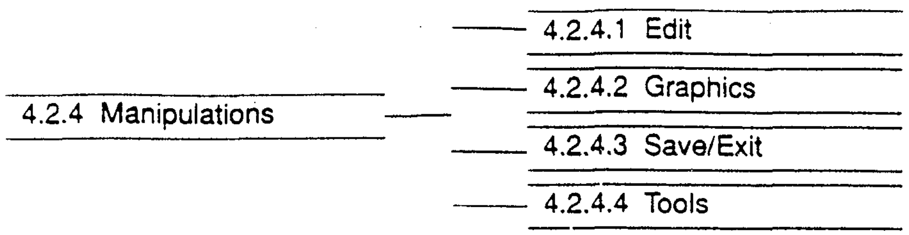
d. The lower limit for resizing a window should be the size of the window title.

RATIONALE: Users should be able to identify a window of any size.

4.2.3.5 TRANSFER OF DATA BETWEEN WINDOWS

a. Data should be transferred between windows with the use of the cut and paste commands (see subparagraphs 4.2.4.1.3 – 4.2.4.1.5 for a detailed description of cutting and pasting).

4.2.4 MANIPULATING INFORMATION IN A DISPLAY



4.2.4 Manipulations

4.2.4.1 Edit

4.2.4.2 Graphics

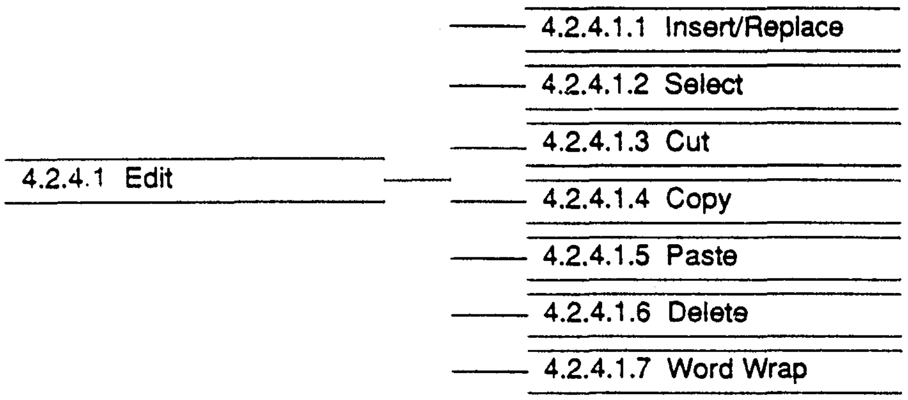
4.2.4.3 Save/Exit

4.2.4.4 Tools

OVERVIEW OF SUBPARAGRAPH 4.2.4

DEFINITIONS AND DESCRIPTIONS: Manipulating information in a display includes selecting information, deleting, moving information either directly or by cutting and pasting, copying, and saving information. The guidelines address various types of display formats: text, graphic, and tabular. See subparagraph 4.2.8.1.1.1.2 for information on the interactive dialogues specific to editing.

4.2.4.1 EDITING



4.2.4.1 Edit

4.2.4.1.1 Insert/Replace

4.2.4.1.2 Select

4.2.4.1.3 Cut

4.2.4.1.4 Copy

4.2.4.1.5 Paste

4.2.4.1.6 Delete

4.2.4.1.7 Word Wrap

OVERVIEW OF SUBPARAGRAPH 4.2.4.1

GUIDELINES:

a. The user should be able to edit text, tables, and graphics by multiple methods (e.g., by use of editing commands in a menu, the UIL commands, and command keystrokes). All editing procedures should be consistent in the dialogue structure and syntax, independent of the type of information being edited.

b. When appropriate (e.g., when reading a read/write file), the user should have the ability to change the physical characteristics of text. Example physical characteristics to put under the user's control include font type, size, and capitalization; the ability to change the font style (e.g., by underlining, italicizing, and/or bolding characters or strings of characters); and/or to alter tab position in any part of a text file.

c. A tab system should be available for subparagraph indentation and moving the cursor to a preselected location. The user should be able to set tabs at locations across a display, consistent with the spacing provided by the space bar. The symbols indicating the location of tabs should be invisible to the user by default but should become visible with a single action by the user (for example, by making a screen ruler appear on the display or displaying the tab symbols within the text field).

d. The user should have the ability to change margins for a text file, up to 160 spaces from the left. This capability should include changing margins so that the user cannot view all of the characters in the horizontal line.

RATIONALE: Users may need to have a double page size for the equivalent of a 14 x 17 page.

e. The user should have the ability to shift the text information shown when the user cannot view all of the characters in the horizontal line. This shift should be accomplished with a single action (e.g., by moving a scroll icon on a horizontal scroll bar).

4.2.4.1.1 INSERTING AND REPLACING TEXT

GUIDELINES:

a. By default, the text editor should operate in insert mode. Text should be inserted moving to the right from the cursor location and should wrap to the next line when necessary (see subparagraph 4.2.4.1.7).

b. If text is selected (see subparagraph 4 2.4.1.2), the editor should be in the overstrike mode. Alphanumeric characters typed following selection of a string of characters should replace the selected text.

4.2.4.1.2 SELECTING DATA FOR EDITING

GUIDELINES:

a. Users should be able to select textual, graphical, and tabular information for specific editing functions (e.g., cutting or copying) by multiple methods (e.g., with a cursor control device or step keys), each requiring minimal simple actions to perform the selection.

b. The selected information should be visually distinct from non–selected items (e.g., by means of reverse video or color).

c. The minimum amount of alphanumeric data that users should be able to select is one character.

d. If the selected text, table, or graphics area extends beyond the bottom of the displayed page, the screen should automatically scroll until the user stops selecting.

e. Users should not be able to select non–contiguous blocks of text.

RATIONALE: Cutting and pasting (operations which frequently follow selecting) is ambiguous with non–contiguous blocks, especially with respect to the spatial relation between the two non–contiguous blocks when they are pasted into a text file at a new location or into a new text file.

f. Users should be able to select multiple non–contiguous graphic objects using minimal actions. Operations available for individual graphic objects should also be available for multiple selected objects.

g. Users should be able to unselect textual, tabular, or graphical information (i.e., remove the information from the selected state) with a single action. Unselection should remove the perceptual cue indicating selection.

4.2.4.1.3 CUTTING

DEFINITIONS AND DESCRIPTIONS: This subparagraph addresses the removal of data (in the form of text or graphics) from a displayed data file. Data that are removed by cutting may be replaced later at any specified location(s). In contrast, data that are deleted (see subparagraph 4.2.4.1.6) may be replaced only at the location from which it was deleted by a specific "undo" command. See subparagraph 4.2.4.4 for a discussion of possible tools for cutting data.

GUIDELINES:

a. Users should be able to cut only data that are currently selected (see subparagraph 4.2.4.1.2).

b. Users should be able to cut textual, tabular, or graphical data by multiple methods, each requiring only a single action (e.g., selecting a menu item or by entering a UIL command).

c. Users should be able to place cut data at any location in the current data file or other data files created with the same application (e.g., in another file created using the same word processing software). One implementation that would allow users to accomplish this is the use of a temporary editing buffer into which the system would place cut data.

d. Users should be able to view data after it has been cut prior to pasting the data.

e. Users should have the capability to place cut data at any location in data files created using other applications (e.g., placing data cut from a word processing file into a graphics file or into a UIL file).

f. The cut data should be removed from the text or tabular file which should be reconstituted without a gap where the text was removed. The cursor should remain in the same location as prior to the cut. A graphical file from which a graphical object was cut should be reconstituted to occupy the same amount of space as before the cut, with a gap where the object was removed.

g. Users should be able to cut both graphical objects and areas of a graphical display.

4.2.4.1.4 COPYING

DEFINITIONS AND DESCRIPTIONS: COPYING allows the user to replicate data and place it at any location without disrupting the original data. See subparagraph 4.2.4.4 for a discussion of possible tools for copying data.

C-2.

GUIDELINES:

a. Only data that are currently selected may be copied (see subparagraph 4.2.4.1.2).

b. The user should be able to copy selected information from the display by multiple methods, each requiring only a single action (e.g., selecting a menu item or by inputting a UIL command).

c. Users should be able to place copied data at any location in the current data file or other data files created with the same application (e.g., in another file created using the same word processing software). One implementation that would allow users to accomplish this is the use of a temporary editing buffer into which the system would place copied data.

d. Users should be able to view data after it has been copied prior to pasting.

e. Users should have the capability to place copied data at any location in data files created using other applications (e.g., placing data copied from a word processing file into a graphics file or into a UIL file).

4.2.4.1.5 PASTING

DEFINITIONS AND DESCRIPTIONS: PASTING involves placing data that have been cut or copied into a data file. See subparagraph 4.2.4.4 for a discussion of possible tools related to pasting.

GUIDELINES:

a. The user should be able to paste the data that was most recently cut or copied into a text, tabular, or graphical file by several different methods, each requiring only a single action (e.g., selecting a menu item or by input of a UIL command).

b. The pasted data should be inserted in the text or table in the location immediately before the cursor and in a graphical file at the approximate location of the cursor. At the end of the paste process, the cursor should have the same data following it in the text or table as before the process.

c. Pasting the most recently cut or copied data should have no effect on a users' ability to paste the same data again. That is to say, the user should be able to paste the most recently cut or copied data as many times as he or she chooses. The data that can be pasted is only replaced when new data are cut or copied.

d. The user should be able to paste alphanumeric data cut or copied from a text file or table into a graphical display and graphical data into a text or tabular file.

4.2.4.1.6 DELETING

GUIDELINES:

a. Data should be deleted after being selected or by the use of a backspace function (see subparagraph 4.2.4.1.2).

b. Selected data should be deleted by a single action different from the actions used for copying, cutting, and pasting (e.g., pressing the backspace key).

c. Deleted data should be stored by the system to allow the user to undo the deletion and restore the deleted text. The user should be able to restore the deleted text by a single action.

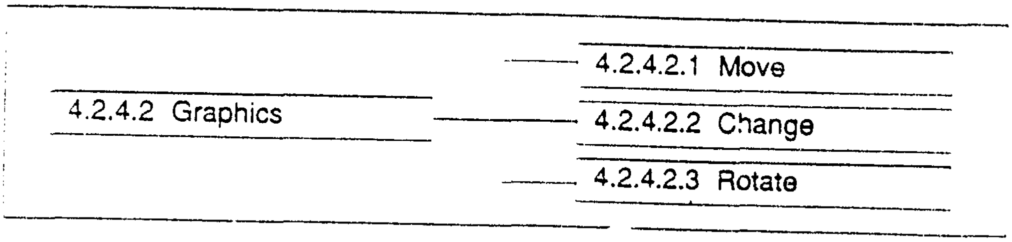
4.2.4.1.7 WORD WRAPPING

DEFINITIONS AND DESCRIPTIONS: Words pasted into a text line will often displace words to the right of the insertion point. Word wrap occurs when words displaced from one line are moved to the next line so as to maintain the continuity of the text.

GUIDELINES:

a. Word wrap should be a default function in the editing application.

4.2.4.2 SPECIAL GRAPHICS EDITING FUNCTIONS



4.2.4.2 Graphics

4.2.4.2.1 Move

4.2.4.2.2 Change

4.2.4.2.3 Rotate

OVERVIEW OF SUBPARAGRAPH 4.2.4.2

4.2.4.2.1 MOVING OBJECTS

GUIDELINES:

a. Only objects that have been selected should be movable within the display.

b. Users should be able to move objects by multiple methods (e.g., direct manipulation or a UIL command).

c. As a user moves an object, an outline of the object should appear to move in a continuous manner; that is to say, the movement should show the outline at intermediate points between the initial and final points in sufficiently fine gradations for the appearance of continuous motion.

4.2.4.2.2 CHANGING SIZE

GUIDELINES:

a. The user should have the ability to resize graphical objects both by direct manipulation of the object and use of the UIL. Types of resizing should include simultaneous resizing of both x and y dimensions and changing only one dimension.

b. The capability to resize an object should include both increases and decreases in size.

4.2.4.2.3 CHANGING ORIENTATION (ROTATION)

DEFINITIONS AND DESCRIPTIONS: ROTATION of the object is defined as moving the object around an imaginary line through the center of the object clockwise or counter–clockwise. The rotation movement is not constrained to the plane of the display.

GUIDELINES:

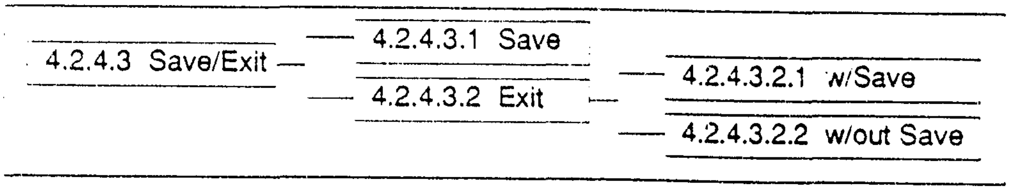
a. Only objects that have been selected should be able to be reoriented.

b. Users should be able to reorient objects by multiple methods (e.g., direct manipulation, selection a menu command, or input of one of a set of UIL commands).

c. The rotation of an object to a new orientation should involve a smooth and continuous motion of an outline of the object (see subparagraph 4.2.4.2.1).

d. Users should be able to rotate objects to the resolution of one degree from one to 180 degrees, clockwise or counterclockwise.

4.2.4.3 SAVING AND/OR EXITING A DATA FILE



OVERVIEW OF SUBPARAGRAPH 4.2.4.3

DEFINITIONS AND DESCRIPTIONS: After finishing their interaction with a data file and periodically during that interaction, users will need to perform one of three actions to the file. During the interaction, they will need to save the data but stay in the data file to continue working on it. After finishing the interaction, users will need to either exit the file and save the new input or exit the file and delete the new input.

4.2.4.3.1 SAVING

GUIDELINES:

a. The user should have the ability to save the data entered into a data file by a single action that will permit the user to continue interacting with that file (e.g., selecting a menu item or by UIL input). This action will replace the previous data stored in the data file with the newly saved data.

b. The system should save a data file automatically at frequent intervals (e.g., every minute or every 20 changes) while being edited. This capability might be used in conditions when the accidental loss of data would be catastrophic or would have a substantial, negative effect on crew productivity.

4.2.4.3.1.1 EXITING A FILE

GUIDELINES:

a. After finishing the interaction with any type of file, the user should be able to stop interacting with the file by a single action (e.g., selecting a menu item or UIL input) without saving the changes to the file. Commands for exiting should be different from those for saving and exiting with a save.

b. Users should be protected against exiting a data file without the opportunity to save the file contents.

4.2.4.3.1.2 EXIT WITH SAVE

GUIDELINES:

a. After finishing the interaction with any type of file, the user should be able to save the data and stop interacting with the file by a single action (e.g., selecting a menu item or UIL input).

b. Users should have the option of invoking an automatic backup function that retains previous versions of data. The specific number of previous versions saved should also be user selectable.

c. To guard against accidentally replacing a data file with an incorrect file, data from a file that has been modified and stored with the "save" or "exit with save" actions should be retrievable with a single action. The previous data should be accessible for at least a brief period of time after the "save" or "exit with save" actions.

d. Any command used to exit with save should be different from the commands for save and for exit without save.

4.2.4.3.1.2.1 EXIT WITHOUT SAVE

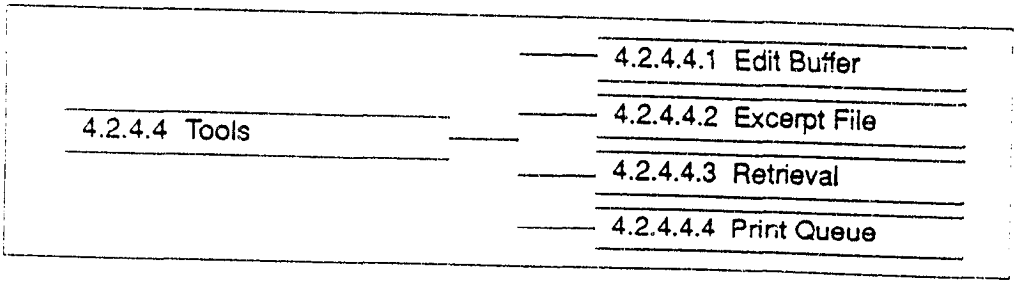
GUIDELINES:

a. Users should be able to exit from any type of data file without saving any new input to the file by a single action (e.g., selecting a menu item or by UIL input).

b. To protect against accidental deletion of new input, the system should require that the user verify that he or she wanted to exit and delete new inputs. For example, after issuing this command, a standard message window might be displayed with a question, "Do you want to exit the file without saving the changes?" The window might also contain boxes or buttons labeled "Yes" and "No" one of which the user should select. The window would be displayed until selection of a button. No other input would be possible during the display of the window. Selection of the "No" button would remove the window and return the user to the data file without any change.

c. As a second level of protection from accidental deletion of new input, data from a file that has been modified by new input should be retrievable with a single action even after exiting with deletion of new input. The modified data file should be accessible for a period of time after the "exit" actions.

4.2.4.4 TOOLS FOR MANIPULATING DATA



OVERVIEW OF SUBPARAGRAPH 4.2.4.4

DEFINITIONS AND DESCRIPTIONS: Certain potential SSFP computer system tools, discussed in this subparagraph, may aid the user in manipulating data, especially in editing and printing data. Although they may affect the user's interactions with the system, the user may not directly interact with the tools per se. The guidelines in this subparagraph focus on those aspects of the tools which would have the greatest affect on the user. In addition, these guidelines should not be read as a requirement for the SSFP systems to use these tools. The guidelines simply suggest what the HCI should be like if these tools are part of a system.

4.2.4.4.1 TEMPORARY EDITING BUFFER

DEFINITIONS AND DESCRIPTIONS: A TEMPORARY EDITING BUFFER (known on some systems as the clipboard) is normally invisible to the user but may be displayed in a window. This buffer is independent of, but able to interface with, all other applications. Its purpose is to hold data temporarily so that it can be moved from one place in a file to another or from one file to another.

GUIDELINES:

a. If selected data is cut or copied from a text file, tabular file, and/or graphics file and placed in a temporary editing buffer, the data should be placed in the buffer automatically, with the only specific action required by the user being the cut or copy action. If a temporary editing buffer is used, data pasted into a text file, tabular file, and/or graphics file should be pasted from that buffer.

b. The contents of the temporary editing buffer should remain intact even if the application from which the contents were taken is closed.

c. The default condition should be that additions to the temporary editing buffer are not cumulative; new data placed in the buffer should replace old data.

d. The user should be able to access the contents of the temporary editing buffer in a window with a single action. Access to the contents of the temporary editing buffer should permit the user to read the contents, but not operate on them.

4.2.4.4.2 EXCERPT FILE

DEFINITIONS AND DESCRIPTIONS: Like the temporary editing buffer, the EXCERPT FILE (known in some systems as the Scrapbook) also allows the user to move data from one location to another. However, the excerpt file permits the user to perform a variety of functions that a buffer does not.

GUIDELINES:

a. Users should have the capability to create multiple excerpt files.

b. The user should be able to paste data into an excerpt file.

c. The user should have the capability to integrate new data with data already in the file. For example, integrating data might include pasting the new data following data already in the file, pasting the new data before data already in the file, and interleaving new data in data already in the file. Each of these capabilities should be available through a single user action.

d. The user should be able to cut or copy data from the Excerpt File and paste it to any other file.

e. The user should be able to save the excerpt file.

4.2.4.4.3 RETRIEVAL BUFFER

DEFINITIONS AND DESCRIPTIONS: The RETRIEVAL BUFFER permits the user to retrieve data after an action that would otherwise have destroyed the data (e.g., saving changes to a file and thereby destroying the old data in that file, or deleting a file).

GUIDELINES:

a. The user should be able to view the contents of the retrieval buffer but not to operate on the contents.

4.2.4.4.4 PRINT QUEUE

DEFINITIONS AND DESCRIPTIONS: The PRINT QUEUE permits the user to print a data file, then to return to working.

GUIDELINES:

a. The user should be able to view a list of the contents of the print queue, but not to operate on the contents, with one exception: The user should be able to delete jobs from the print queue selectively.

4.2.5 MOVEMENT THROUGH A STRUCTURED SEQUENCE OF DISPLAYS

DEFINITIONS AND DESCRIPTIONS: In working with the Space Station Information System, Technical and Management Information System, and Software Support Environment, users (both ground–based and space–based) will have to interact with objects and data files that have a hierarchical or network structure. Such structured objects and files include a menu hierarchy, sets of procedures, and databases. In interacting with these types of objects and files, the users will have to move from one display to another in the hierarchy or network.

GUIDELINES:

a. When hierarchical levels are used to control a process or sequence the number of levels should be minimized (see the current release of NASA–STD–3000, Vol. IV, for current SSFP requirements on movement through hierarchical structures).

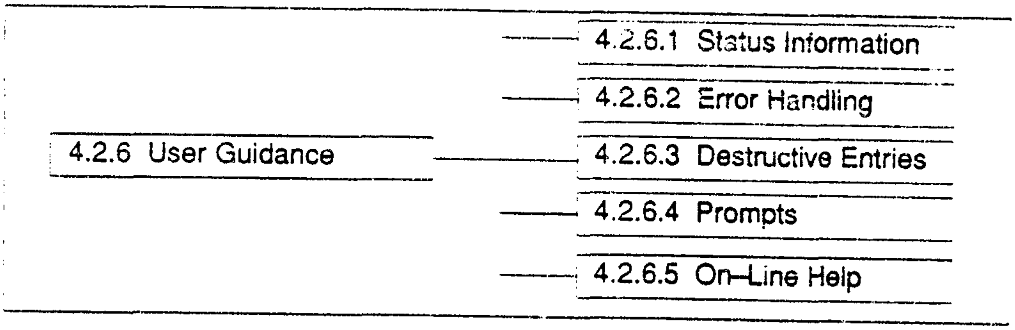
b. Display and input formats should be similar within levels and the system should indicate the current positions within the sequence at all times (see the current release of NASA–STD–3000, Vol. IV, for current SSFP requirements on movement through hierarchical structures).

c. Users should have the capability to skip levels of the hierarchy (see the current release of NASA–STD–3000, Vol. IV, for current SSFP requirements on movement through hierarchical structures).

d. In a hierarchical menu, users should be able to return to the main menu with a single action. (ref. 64, section 3.1.3.34.)

e. The user interface should provide the user with information and actions that he or she needs to navigate in a structured data file or object. For example, a flowchart might be provided to show the user, upon request, his or her position within such hierarchies as menus, a hierarchy of operations for payloads or Station management, and hierarchical data bases. The flowchart might be contained in a window separate from the application that the flowchart represents. In this example system, movement from any position in the hierarchy to any other position might be accomplished by using the cursor control device to select the desired position on the flowchart. In addition, the flowchart box for a user's current position in the hierarchy would be highlighted. An alternative example would be if displays in a hierarchy were identified by a convention, such as application, operation, location in the hierarchy. An identification code for each display might then consist of abbreviations for the application and operation and numbers for the location in the hierarchy. Then, users would be able to access a display in a hierarchy by inputting a command identifying the display and directing the system to go to it.

4.2.6 USER GUIDANCE



4.2.6 User Guidance

4.2.6.1 Status Information

4.2.6.2 Error Handling

4.2.6.3 Destructive Entries

4.2.6.4 Prompts

4.2.6.5 On–Line Help

OVERVIEW OF SUBPARAGRAPH 4.2.6

DEFINITIONS AND DESCRIPTIONS: Users sometimes need for the system with which they are working to guide them through their task. User guidance does not simply provide feedback about user errors but also includes all feedback from the system that indicates the actions that are available to the user.

4.2.6.1 STATUS INFORMATION

GUIDELINES:

a. Each status message should be given precedence according to its importance (see the current release of NASA–STD–3000, Vol. IV, for current SSFP requirements on precedence of status messages). In addition, status messages that inform the user of an impending shut down of the system should interrupt the user's task–related interactions with the system. Messages that inform the user of an important and timely but noncatastrophic event for the system should have a unique signal associated with it but will not interrupt the user's on–going interaction. Messages that provide the user with status information that is not timely should also be signalled by a unique signal that will not interrupt the user's on–going interaction.

b. The system should automatically provide users with information about the current system status as it affects their work.

c. Status messages should be timestamped and users should have the capability to view messages by timestamp.

4.2.6.1.1 TYPES OF STATUS INFORMATION

4.2.6.1.1.1 LOGON

GUIDELINES:

a. The prompt for logging on should be displayed automatically at a workstation connected to an SSFP computer system as soon as the workstation is activated. (ref. 64, section 4.1.1.) The logon procedure should not be necessary for crewmembers on board Freedom Station.

b. If a user cannot log on to the system because the system is unavailable, a message should be displayed at all SSFP computer system workstations. This message should inform the user that he or she cannot log on to the system and the logon will become available at an estimated time. The time estimate should be updated to reflect up–to–date estimates of system availability.

c. If a user cannot log on due to use of an incorrect LOGON procedure, the system should provide a message, within system security restrictions, explaining that the user's LOGON procedure was unsuccessful and why the procedure failed (e.g., incorrect LOGON sequence, expired password).

4.2.6.1.1.2 OPERATIONAL MODE

GUIDELINES:

a. The system should inform the user of the current operational mode when the mode might affect the user's actions. (The relevant operational modes are not yet defined.) For example, when the system is processing data, a message like "Processing file ______________" might be displayed in the message window.

b. The system should also provide information about nonoperational mode status, e.g., "System down" and "Keyboard Locked." In addition, when the user attempts inputs to a nonoperational system, an auditory signal (e.g., a beep) should accompany each input as feedback to the user. (ref. 64, section 4.1.5.)

4.2.6.1.1.3 OTHER USERS

GUIDELINES:

a. Information should be available to the user about the number, identity, or status of other system users, within established system security restrictions.

b. When the system provides the user with a list of the names of other users, it should also provide a means by which the user could contact them on request to facilitate communications (e.g., an electronic mail address).

c. The system should acknowledge that a specific user is logged on to the system when that information is specifically requested by another user.

4.2.6.1.1.4 CHANGES IN SYSTEM DESIGN OR OPERATIONS

GUIDELINES:

a. The system should inform users about system design or operation changes only in those aspects that may affect the users' interactions with the system (e.g., response time, on–line user guidance, commands, or menus) at the time of logging on or when the information becomes available to the system. The message should contain only the following basic information required by the user: a description of the system change, directives for user action, and the consequences of failing to follow the directives. An example system message might read "Disk problems: Save data and log–off immediately or lose data."

b. The user guidance and Help functions should be updated to reflect changes in system design or operations that affect the users' interactions with the system.

4.2.6.1.1.5 PARAMETER VALUES AND OBJECT ATTRIBUTES

The user should have the capability to request relevant information related to the current operation, including default value parameters, current value parameters, and object attributes.

4.2.6.1.2 TECHNIQUES FOR PROVIDING STATUS INFORMATION

4.2.6.1.2.1 VISUAL

GUIDELINES:

a. Status information should be provided in a specific message window placed in a consistent screen location.

b. The messages displayed in the window should be in understandable standard English text free from computer jargon. The use of standard acronyms and abbreviations are acceptable (see subparagraph 4.1.1.1.2.2 for detailed guidelines on acronyms and abbreviations).

4.2.6.3 CONFIRMATION OF DESTRUCTIVE ENTRIES

GUIDELINES:

a. Users should have to confirm that they want to perform a destructive command before the system will execute it. This guideline applies to replacing or destroying any data file, any disk, or other stored data. The guideline does not apply to editing of words, numbers, or text lines within a document or data form or to editing of graphical elements within a graphical data file.

b. The co'ifirmation message should be simple, positive, and direct. For example, the message might be: "This command will destroy (replace) data file XX. Do you wish to destroy (replace) this file?" and might be accompanied by boxes or buttons labelled YES and NO which the user could select with the cursor control device. As an example of a direct, positive confirmation, an "EXIT" command may be acknowledged wiuh "CONFIRM EXIT" but not with "ABORT EXIT."

c. YES and NO responses to confirmation messages consistent throughout the system. For example, it is not appropriate to have respond to the confirmation message "Quit without saving chang n one application and "Save data before exit?" in another.

4.2.6.4 PROMPTS

GUIDELINES:

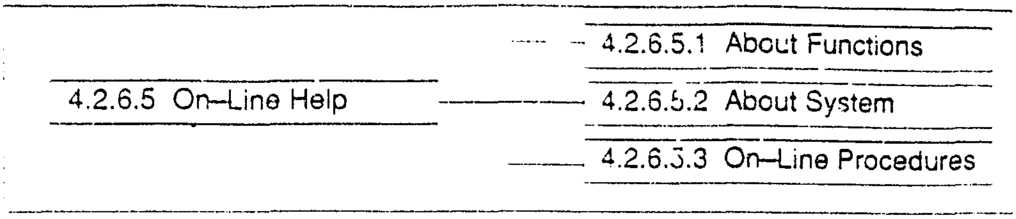
a. The system should provide prompts for the next sequential rigid test procedure in a task sequence. For example, if a STEP key should be pressed after the completion of a step in a task, the message area should display "Press STEP key to continue." (ref. 64, section 4.4.5.)

b. The system should provide prompts for entering data (e.g., in a data form) or UIL command inputs. (ref. 64, section 4.4.7.)

c. The location for prompts for data in a data form should be at the location of the required data. The location for prompts for data or commands on other types of displays should either be at the location of the data or command or in the message area.

d. If a prompt requires an input, as many features of that input as possible should be specified as part of the prompt. For example, the required input might be indicated by the use of a colon followed by underlining that extends for as many spaces as the input, as well as any necessary punctuation. For example, a prompt requesting the input of a date should be, "Please list the date of Event A (DD/MM/YY): __/__/__."

4.2.6.5 ON–LINE INSTRUCTION AND HELP



4.2.6.5 On–Line Help

4.2.6.5.1 About Functions

4.2.6.5.2 About System

4.2.6.5.3 On–Line Procedures

OVERVIEW OF SUBPARAGRAPH 4.2.6.5

GUIDELINES:

a. The design of the Help function should be consistent with system security restrictions.

b. The system should provide help to user. h on request and, in certain conditions, automatically. A condition for automatic help might be frequent errors in a specific interaction with the system. However, users should be able to limit automatically–presented help displays with a single action.

c. Users should have multiple methods of requesting help. For example, a user might select Help in a pull–down menu, type a "Help" command, and/or press a Help Function Key.

d. Users should be able to access Help at any point in their interaction with the system, including prior to logging on to the system, within the system security guidelines. That is to say, users should be able to access and read from the Help database as a form of on–line instruction about system functions and operating procedures.

e. A Help request from the user should elicit a task–specific and context–sensitive response from the system. For example, if a user were performing a data analysis task and requested Help, the system should provide information specific to the data analytic techniques available to the user.

f. If the task in which the user is engaged cannot be identified unambiguously, the system should query the user to specify the data, message, menu item, or command that resulted in the request for help.

g. The Help function should provide summary information about the requested topic as the initial level of Help. At the user's request the help function should provide more specific information about the topic. The user should select the specific area of additional information by selecting from a set of Help items displayed to the user. If the user must read through several levels of Help, the system should provide contextual information to aid in navigating through the Help Menus (see the subparagraph 4.2.5 for detailed information on Movement through a Structured Sequence of Displays).

h. The Help function should be displayed to the user in text and/or annotated graphics (e.g., schematics or flowcharts), as is appropriate to the topic on which help has been requested. The text should be in simple sentence structure with proper punctuation. The text should be concise and should directly address the topic without digression.

4.2.6.5.1 INFORMATION ABOUT FUNCTIONS AND SYNTAX OF COMMANDS

GUIDELINES:

a. At the user's request, the Help function should provide the user with an index of UIL commands that lists, at a minimum, the full UIL command, any associated keystroke command or abbreviation, and any associated menu–based command. The command index should provide the user with information about the proper syntax and command wording and with common synonyms. To aid in finding the correct command, the user should be able to search the list by synonyms, as well as by correct commands. For example, the structure of the index might be as a comprehensive index of commands that is organized according to system operations and functions or an index of commands that is specific to the user's current task.

4.2.6.5.2 INFORMATION ABOUT SYSTEM OPERATIONS

GUIDELINES:

a. The Help function should include information about system design and system operations that affect the users' interactions with the system. For example, the Help information about system operations might include listings of available system operations and instructions about the use of those operations.

4.2.6.5.3 ON–LINE DOCUMENTATION AND PROCEDURES

GUIDELINES:

a. The user should be able to access on–line documentation and descriptions of procedures. For example, titles of documents and procedural descriptions might be listed in a data file from which the user could access them directly.

b. Consideration should be given to displaying documentation and procedural data in a flowchart format. Evidence exists which suggests that flowcharts are more easily scanned for information (ref. 18).

c. When procedures must be displayed in a text format, the system should provide the user with contextual information indicating the current procedural step (ref. 18).

4.2.7 COMMUNICATION BETWEEN USERS

GUIDELINES:

a. A user should be able to send messages to other users on the same system and to read messages that have been received.

4.2.7.1 SENDING MESSAGES

GUIDELINES:

a. A user should be able to communicate with other users without exiting an application in which he or she is working.

b. A user should be able to communicate interactively with other users who are currently using the same system and to send a message to users who have accounts on that system but are not currently using the system.

4.2.6.1.2.2 AUDITORY

GUIDELINES:

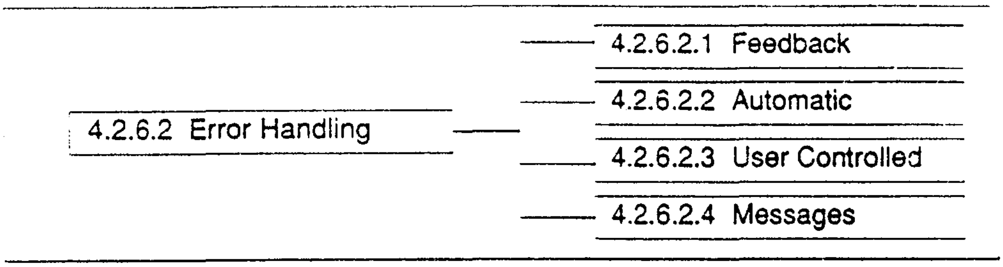
a. Status messages should be accompanied by a consistent auditory signal to alert the user that a message has been displayed.

b. Auditory messages should never be the primary source of information. They should always be redundant with visually displayed messages.

c. Verbal status messages may be given by a voice production system in a situation where the user will not be likely to look at the display. A vocal status message should be repeated at the user's request and should be redundant with a visually displayed message.

d. The user should have the ability to mute the voice system.

4.2.6.2 ERROR HANDLING



OVERVIEW OF SUBPARAGRAPH 4.2.6.2

GUIDELINES:

a. The SSFP computer systems should handle user errors by providing feedback to the user to permit him or her to identify and correct the error and by automatically providing information that will be an aid in correcting the error.

4.2.6.2.1 ERROR FEEDBACK

GUIDELINES:

a. When a user makes an incorrect command entry, the system should notify the user by a message within 2 seconds after the error. (ref. 64; section 3.5.1, p. 366.) The message should be in understandable English and should not be caustic or accusatory. For example, if a user incorrectly input alphabetic characters for coordinates x, y, and z in a proximity operations task, the error message might read "Non–numeric input given. Numeric input is required."

b. The error messages should be specific, informative, and brief. The message should consist of a description of the error and directives for actions that will correct the error. In addition, if necessary, the error message should also include a description of the consequences of failing to correct the error. The user should be able to access "Help" or "Query" to provide a more detailed message. (ref. 60.)

c. Generally, error numbers are of little help to the user. Accordingly, error numbers should be provided only for purposes of documentation. The user should not have to refer to the external documents in order to interpret the error message (ref. 60).

d. Error messages should be placed on the display at the point of the error and/or in a designated, consistent area of the display. (ref. 60.)

e. The system should only require correction of that subpart of the input that is in error. The original input should be presented to the user for editing or, when helpful, prompts should be given to the user for the subpart of the input that is in error. For example, after the incorrect input of alphabetic characters for coordinates, the system might respond with the following:

Coordinate x: ?

Coordinate y: ?

Coordinate z: ?.

f. After the user has corrected the error, the system should remove the error message from the display.

4.2.6.2.2 AUTOMATED AIDS TO ERROR CORRECTION

GUIDELINES:

a. As an automated aid to error correction, the system should provide a list of permissible commands, or commands predicting what the user is attempting. (ref. 60.)

4.2.6.2.3 USER CONTROLLED ERROR CORRECTION

GUIDELINES:

a. The user should be able to edit any part of a command or the entire command. (ref. 64, section 3.5.3.)

b. If an error is detected in a stacked series of command entries, the system should execute to the point of error. The user should be notified and provided with appropriate guidance to permit correction, completion, or cancellation of the command that is in error. (ref. 64, sections 3.5.4–5.)

c. The system should require a distinctive confirmation action for destructive entries that warns the user of potential data loss. Whenever possible, the user should be allowed to reverse such a control action. (ref. 64, sections 3.5.10, 3.5.7.)

4.2.6.2.4 CAUTIONARY MESSAGES

GUIDELINES:

a. When the user's input appears to violate the normal range of entries for a data category or a command, the system should display a message asking the user to confirm or change the input. The message should describe the input that appears to violate the conditions and the user's possible actions. For example, if the input during a Life Sciences experiment on exercise showed heart rate at 350 beats per minute, the message might be as follows: "Heart rate of 350 bpm is outside the normal range; please confirm or change entry." Note that this type of message differs from an error message which would be given following any impossible data entry (e.g., a negative number or an alphabetic character for heart rate data) in that it permits the user to confirm the input.

c. A user should be able to communicate with and/or send messages to multiple users simultaneously.

d. In general, a user should create messages according to the guidelines for input and editing of text (see subparagraphs 4.1.1.2.2.1 and 4.2.4.1). However, users should be able to input unformatted messages (ref. 64, section 5.1.2), and if a message requires a standard format, the system should provide the user with the format.

e. A user should be able to incorporate an existing file into a message. (Ref. 64, section 5.1.11.)

f. Users should be able to save or delete messages that they have created prior to or after sending.

g. A user should have the ability to print created messages, whether sent or unsent.

h. The user should be able to send the message with a single action (e.g., selecting a menu command or inputting a command with the command language). The user should be able to command the system to hold a message for a prescribed period of time or to hold the message for release conditional on an additional action by the user.

i. The user should have access to a log of all message transactions, both as a sender and as a receiver.

j. The system should aid users in communicating. For example, the system might be able to display a data file with names and addresses from which the user could select names for communicating with or sending mail to.

k. The system should transmit messages of any length between users (ref. 64, section 5.1.12), and should also permit users to select and transmit parts of messages.

4.2.7.2 RECEIVING MESSAGES

GUIDELINES:

a. Users should be notified of messages that they have received. For example, users might be notified of all recent messages when they log–on to the system or at specified times.

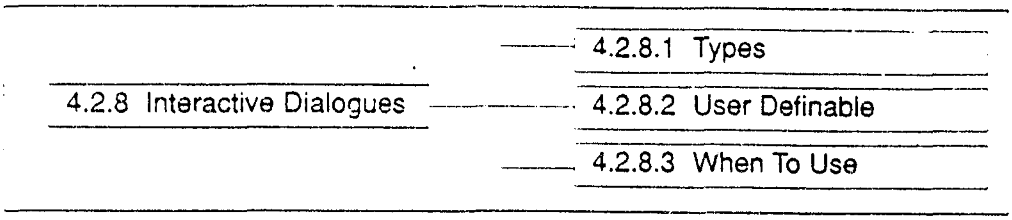
b. In general, notification of the receipt of messages while a user is logged–on to the system should not disrupt the user. Exceptions include announcements of an emergency by the Space Station Commander or announcements of critical importance to users by the System Manager.

c. The user should be able to select a specific message to read with a single action. The user should be able to select messages from an ordered queue; selection of any message in the queue should be permitted. At the minimum, the queue should list the message title, sender, date, and time of receipt. The default order for the queue should be according to the date and time of receipt. The user should have the ability to reorder the messages according to title, sender, or any other identifying characteristic.

d. The display of messages should follow the format for text (see subparagraphs 4.1.1.2.2.1 and 4.2.4.1).

e. Users should be able to delete or save messages either before or after reading them. To protect the user from accidentally destroying a message by deleting it, users should be able to retrieve the message with a single action.

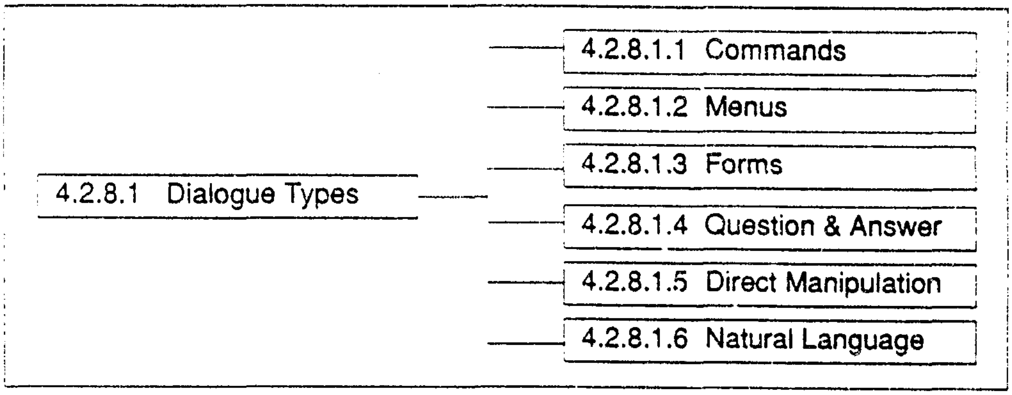
4.2.3 INTERACTIVE DIALOGUES



OVERVIEW OF SUBPARAGRAPH 4.2.8

DEFINITIONS AND DESCRIPTIONS: DIALOGUES are the means by which the human provides commands to the computer and by which the computer provides the human with messages, answers to queries, status information, and help.

4.2.8.1 DIALOGUE TYPES


4.2.8.1 Dialogue Types
4.2.8.1.1 Commands
4.2.8.1.2 Menus
4.2.8.1.3 Forms
4.2.8.1.4 Question & Answer
4.2.8.1.5 Direct Manipulation
4.2.8.1.6 Natural Language

OVERVIEW OF SUBPARAGRAPH 4.2.8.1

DEFINITIONS AND DESCRIPTIONS: The design of each of the following dialogue techniques is described by the recommended structure and representation, input and output, and feedback of the technique. Structure refers to the underlying relationships among components of a dialogue. Representation refers to that aspect of the interface visually apparent to the user and global in nature (basic display structures, etc., see also subparagraph 4.1.1.1). The explicit actions of the user and associated display changes are described under Input and Output. Appropriate completion, error, and mode messages are included within Feedback.

GUIDELINES:

a. To the greatest degree possible, the user should be able to interact with and input commands to the SSFP computer systems using multiple dialogue types. The systems should allow a user to switch among dialogue types within a task sequence, although a given command string should be in a single dialogue type. For example, use of the UIL to accomplish certain commands should not preclude the use of menus, function keys, direct manipulation, or any other accepted dialogue technique to accomplish other commands, even within the same task transaction. (e.g., the UIL should provide a means for facilitating rapid access to the various applications, devices, and features of a Space Station Freedom Computer System. At a minimum, major applications of the system should be available via a simple command.) The user should make the choice of the appropriate dialogue technique at every point in the transaction, based on the his or her needs and the characteristics of the task.

b. The different dialogue types should share a common framework, so that the user can learn the basic dialogue concepts — objects, the actions that can be performed on an object, and the attributes of the objects and actions — and transfer those concepts between dialogue types.

c. The human–computer dialogues should allow users to execute data manipulations (e.g., data analysis, integration of data from remote locations) in terms of the functions to be performed without concern for internal computer data processing, storage, or retrieval mechanisms (e.g., specifying the location of or pathway to the data). The dialogue techniques, however, should not prevent users from specifying pathways when such specification is preferred by the user, for example, when multiple copies of a file with the same name reside in different devices.

d. In order to ensure that all user–computer dialogue types are easy for the user to learn and remember, even with infrequent usage, the lexicon, semantics and syntax of the dialogue technique (i.e., the set of elements of the dialogue, the meanings associated with each element, and the structure of the dialogue, respectively) should be consistent with user mental models, stereotypes and expectations. Design and selection of dialogue elements and structure should facilitate grouping and arrangement of UIL features.

4.2.8.1.1 COMMANDS

DEFINITIONS AND DESCRIPTIONS: COMMANDS are user–initiated messages to the system used to specify desired functions. There are two main types of commands: the UIL, which consists of word strings in a syntactic structure that may be keyed (or, potentially, spoken) onto a command line, and keystroke.commands, which are a limited number of nonlinguistic keystrokes that define UIL commands. The keystrokes are often initiated by the simultaneous press of a key that signals a keystroke command and the first letter of a one word command. Another version of the keystroke command is the function key.

4.2.8.1.1.1 USER INTERFACE LANGUAGE

GUIDELINES:

a. The UIL should be the prototypic dialogue type in that knowledge of the meaning of UIL terms and syntax should transfer to all other dialogue types to the greatest extent possible.

4.2.8.1.1.1.1 STRUCTURE AND REPRESENTATION

GUIDELINES:

a. The functionality, design, and operation of the UIL should be consistent throughout the SSFP. This consistency should include UIL lexicon, syntax and semantics (see subparagraph 4.2.8.1.1.1.1g), and the location, design and operation of the UIL window (see subparagraph 4.2.3 on Window Management).

b. The terms in the UIL should describe the following UIL elements: actions, objects, prepositions, and the attributes of actions or objects.

c. The meaning of terms in the UIL should correspond to a familiar English language meaning of that word.

d. Terminology used to designate elements (i.e., objects, actions, and attribute names) of the UIL should be unique and unambiguous, emphasize distinctive differences between elements, and be maximally meaningful to users in terms of the element's function (e.g., consequences of command execution).

e. Users should be able to use a synonym for an acceptable UIL term with minimal disruption of performance. Two approaches to the use of synonyms might be implemented; however, each approach has advantages and disadvantages.

The system might recognize and accept all probable synonyms for each keyword defined in the command language. Acceptable synonyms would be limited by the requirements of uniqueness and unambiguousness. An advantage to this approach is that the user would have the ability to enter a wide variety of terms and have the system recognize and accept them, thereby reducing errors and saving time. However, a disadvantage is that users show little consistency in their use of synonyms, either from time to time for a single person or across groups of people. Accordingly, acceptance of synonyms may create a tremendous burden for the system. In addition, accepting synonyms may increase the time required to learn the UIL.

The system might recognize all probable synonyms for each keyword in the UIL; but, rather than accepting the synonym, the system would provide the user with an error message which proposed the correct UIL term and allowed the user to insert the correct term in the command. An advantage of this approach is that it may be less burdensome on the system, which would only have to accept one unique UIL term for a given UIL element (although the system would still have to recognize the synonyms). A second advantage to this approach is that it would be more likely to help the user learn the UIL. The major disadvantage is that error trials would be relatively time consuming.

f. Within the constraints of system safety and functionality, frequently used, critical, or time constrained UIL command statements should require minimal keying. For example, minimal keying may be accomplished through the use of function keys, multiple keypresses, macros, or abbreviated commands (see subparagraph 4.1.1.1.2.2 for Abbreviations and Acronyms in Titles, subparagraph 4.2.8.1.1.2, Command Keystrokes and Function Keys, and subparagraph 4.2.8.2, User Definable Dialogue Components).

g. UIL terminology should be consistent in terms of lexicon, semantics, syntax, and consequences throughout the SSFP, irrespective of specific transactions, tasks, locations, and applications (ref. 64, section 3.1.5.7; ref. 3, p. 408). Interpretation of UIL commands by the system should not be affected by superficial characteristics of command statement, such as letter case or spacing.

h. Interpretation of UIL commands by the system should not be affected by superficial characteristics of command statements such as letter case or spacing.

i. The UIL should have a consistent syntax across commands and across applications. The syntax should resemble the structure of English as closely as possible.

j. In order to ensure that the UIL is easy for the user to learn and remember, even with infrequent usage, the lexicon, semantics, and syntax of the UIL should be consistent with user mental models, stereotypes, and expectations. Design and selection of UIL terminology and syntax should facilitate grouping and arrangement of UIL features.

k. Users should be able to use alternative forms of the UIL syntax with minimal disruption of performance. As with synonyms, the following two approaches have important advantages and disadvantages:

— The system should recognize and accept probable alternative forms of UIL syntax (ref. 64, p. 261).

— The system should recognize but not accept alternative forms of syntax, but should provide a user who has used the incorrect syntax with the correct UIL syntax and allow the user to replace the incorrect with the correct.

The relative disadvantages and advantages of the two approaches are similar to those described in the discussion on synonyms.

l. Partial syntactic structures should also be permitted as required. For example, if a user wanted to save inputs to a data file in which he or she was working, the command line could consist of only the action term, "Save." In this example, modifiers for the action are not applicable. The preposition term would be assumed by the SSFP computer systems to be "To," one object term would be assumed to be the data that had been created and its modifier (current), and the second object term would be assumed to be the data file in which the user was working. Thus, in this example, the Command "Save;" would represent "Save data (current) to data file 1;".

m. Special symbols and punctuation should not be used to distinguish related, potentially ambiguous command statements. For example, requiring the user to issue the command "@PRINT" to print one type of file and "$PRINT" to print another type of file is not appropriate.

n. Spaces should serve as delimiters between terms in a command. Any number of spaces between terms should be accepted as delimiters.

o. Two distinct command delimiters (other than spaces or periods) should be used to indicate the syntactic termination of a command (the command separator) and the intent to submit the command to the system (the command evaluator). These command delimiters each should have a unique function in the UIL. The command delimiters' functions should not conflict with any functions that they have in common English usage. For example, a semi–colon is commonly used as a command separator and a carriage return alone as the command evaluator. However, the UIL should minimize requirements for the user to enter superfluous characters, such as arbitrary field delimiters.

p. Where possible, UIL commands that extend beyond one line should word wrap to the next line and should not divide words. In addition, users should have the ability to move to the next line by the use of an evaluator escape (e.g., a backlash) and the command evaluator.

q. The UIL should permit the user to query the system. The query will be especially useful during tasks in which information retrieval is unpredictable, such as planning or analysis tasks. All of the guidelines for the UIL listed in this subparagraph apply to queries.

r. The user should maintain control over the command evaluator. The system should not initiate command statement execution as a side effect of any other action.

4.2.8.1.1.1.2 INPUT AND OUTPUT

GUIDELINES:

a. In general, the UIL will be input in much the same way as text (see subparagraph 4.1.1.2.2.1.). Editing of commands in the command area should follow the same rules as text editing.

b. Commands should be typed in the command area (see subparagraph 4.1.1.1.6.2).

c. The UIL should provide a limited capability for the user to create, name, store, retrieve, execute, and modify command sequences (i.e., stacks and macros). Entering the macro–creation mode should produce an immediate and unambiguous signal to the user to differentiate that mode from the immediate execution mode. (ref. 3, p. 44; ref. 64, section 3.1.5.1.3.) See subparagraph 4.2.8.2 for additional guidelines about macros.

d. The UIL should provide a mechanism whereby the user can designate which task/application is being addressed (see subparagraph 4.2.3, Interactions With Windows and Window Management).

e. Within the constraints of system safety and functionality, the UIL should provide a flexible means of sequence control so that users can accomplish necessary transactions involving data entry, display, and transmission, or can obtain guidance as needed in connection with any transactions.

f. Within the constraints of system safety and functionality, users should maintain initiative and control during transactions, including defining the sequence of transactions (i.e., arbitrary computer processing constraints should not dictate sequence control).

g. The UIL should permit users to enter abbreviated forms of command statements. Abbreviated forms may consist of truncated command action keywords, function keys, and multiple keypresses (see also subparagraphs 4.1.1.2.2.2 and 4.2.8.1.1.2).

h. Users should be able to translate UIL commands and macros into their equivalent in other relevant dialogue types (e.g., menus, direct manipulation interface) to the greatest extent possible and with minimal effort. Likewise, users should be able to translate commands and macros from other dialogue types into their UIL equivalents. Accordingly, the system should aid the user in translating from UIL to other dialogue types and vice versa. For example, commands or macros created under the direct manipulation interface should be displayable upon request in the terminology of the UIL so that they may be read and modified by the user.

i. The user should have access to previously–issued commands within the current transaction which may be re–executed as they are or edited.

j. When there are errors in stacked entries, the entries should be processed until the error is detected. An error message (see subparagraph 4.2.6.2) should then appear. After correction, the computer should process all commands until another error is found or until all commands are completed.

4.2.8.1.1.1.3 FEEDBACK

GUIDELINES:

a. If the completion of the action commanded has a result that is visible to the user, feedback should be communicated by the completion of the commanded action. If the completion of the command has no visible result, feedback should be communicated by a message in the Message Area.

b. The system should acknowledge a command that cannot be completed by a message indicating non–completion of the command and an appropriate error message (see subparagraph 4.2.6.2).

c. The system should acknowledge a command not permitted by the system with a message indicating non–completion of the command and an appropriate error message (see subparagraph 4.2.6.2).

d. The system should permit users to correct errors in command language rather than simply rejecting the command. The command in error should be displayed so that the user can correct and re–enter it. Where possible, the incorrect portion of the command should be highlighted. Capabilities available for command revision within the UIL should be consistent in terms of user actions with those used in other text editing functions within the system (see subparagraph 4.2.4.1 Editing). However, the user should have the ability to replace the command with any other command. (ref. 64, section 3.1.5.24.)

e. The system should acknowledge a potentially hazardous or destructive command as specified in subparagraph 4.2.6.3 (ref. 64, section 3.1.5.25).

f. During immediate execution mode, if the execution of a command statement will have adverse consequences, such as permanent loss of data, the UIL should request user confirmation prior to execution. The request should include a statement as to the exact nature of the consequences of executing the command statement.

4.2.8.1.1.2 COMMAND KEYSTROKES AND FUNCTION KEYS

DEFINITIONS AND DESCRIPTIONS: The guidelines for Command Language should be applied to Command Keystrokes and Function Keys where applicable. Guidelines specific to command keystrokes and function keys are listed below. (See subparagraph 4.3.2 for guidelines related to input from keystroke commands.)

4.2.8.1.1.2.1 STRUCTURE AND REPRESENTATION

GUIDELINES:

a. Keystroke commands should be limited to the simultaneous press of two keys, one that identifies the action as a keystroke command and one that defines the command.

b. There should be a limited number of function keys to perform highly frequent actions.

c. Function keys should command an action with a single press and should not require any preceding keystroke (e.g., pressing the CONTROL key).

d. Each function key should be labelled clearly and uniquely to describe its function.

e. Pressing a function key generally should result in a single action that does not change with repeated key presses.

f. Keyboard commands should be consistent across applications.

4.2.8.1.1.2.2 FEEDBACK

GUIDELINES:

a. Pressing a function key in a sequence of key presses unrelated to the function should result in a message asking the user if he or she really meant to select that function and should not result in the action normally produced by the key until the user responds positively to the question.

b. Feedback should be communicated primarily by the simple completion of the command.

4.2.8.1.2 MENUS

DEFINITIONS AND DESCRIPTIONS: Menu selection provides a user–initiated transaction sequence that permits the user to select a control option or set of control options from a display of available options. Menus can be implemented in several different types, including fixed, pop–up, and pull–down. See subparagraph 4.1.1.1.2.6.3 for definitions of various types of menus and guidelines concerned with the visual display of menus.

4.2.8.1.2.1 STRUCTURE AND REPRESENTATION

GUIDELINES:

a. Menus should be designed so that the function of the menu is evident to the user.

b. Pull–down and pop–up menus should be activated by only a specific user action that requests the display of the menu (e.g., a press on the selection button). Menus should not appear simply because the cursor has passed over the menu title.

c. For all types of menus, menu items that are available to be selected should be highlighted whenever the cursor passes over them and the selection button is down. As soon as the cursor passes outside the boundaries of the menu item the item should return to its normal state. Unavailable options should not highlight when the cursor passes over them.

d. When a pull–down or pop–up menu item has been selected, the menu should revert to its hidden state as the selected command is carried out.

e. Menus should have a limited number of items in breadth (e.g., number of menus in a menu hierarchy, number of menu categories in a menu bar, or number of pop–up menus) and in depth (e.g., number of items per menu or, in menu bars, per menu category). Moderate menu breadth and depth should be facilitated by the use of a hierarchical menu structure whereby the selection of items from one menu (the parent) activates a second menu (the child) with further options (e.g, modifiers for the item selected from the previous menu). The parent menu should remain visible during the selection of the child menu. The number of levels in the hierarchy should be limited (for example, to no more than three).

f. If hierarchical branching is used, only one user action should be required to either move back to the previous menu or return to the main menu.

g. If hierarchical branching is used, an indicator that selection of a menu item will activate a branch to a subordinate list of menu items should be provided. For example, an arrow pointing horizontally to the right and positioned at the right side of the menu item might be included.

h. The user should be able to access a visual representation of his or her path through a hierarchy of menus. For example, if a user progresses through a series of permanent menus, an icon showing the previous menus and current menus, as well as menu selections, might be displayed. If a user progresses through a series of pull–down menus, the previous menus might remain displayed with the selected item highlighted and the association between that item and the subsequent menu would be represented by a close spatial relation (e.g., a walking menu).

i. Items in a menu may be ordered from top to bottom in the menu according to the following three various rules: the most frequently chosen items (ref. 25), the temporal sequence of the actions named by the items, and task–based needs.

j. When the same menus appear in different applications, items in the menus should have the same functions, and functions in the different menus should be named consistently (ref. 64, section 3.1.3.19).

k. For menus on which users will be highly trained and for which spatial location will not be a salient cue for the identity of a menu item, when a menu item is unavailable, it should not be displayed in the menu (ref. 25). In other cases, especially when the menus will need to be learned "on the job" and/or location is a salient cue for the identity of the item, when a menu item is unavailable, it should be kept in the menu but made perceptibly different from the available items.

l. Users should be able to group a series of menu–based commands together and define them as a "macro" command.

m. Users should have the option, with menu choices, of stacking command entries (i.e., of issuing command entries faster than the system can process them). (ref. 64, section 3.1.3.36.)

n. When keyboard equivalent operation is possible, indication of the keys which activate the menu item should be made in close spatial proximity to the menu item and should be clearly associated with it. (See subparagraph 4.2.8.1.1.2 for guidelines on command keystrokes.)

o. If menu items are selectable via activation of programmable function keys, the arrangement of the menu list should be compatible with the arrangement of the keys to the greatest degree possible.

p. When a function key has been used to select a menu list item the selection should be indicated in the displayed item list in the same way that a pointing device entry is indicated, as described in subparagraph 4.1.1.1.6.3

q. If menu items are selectable by keyboard entry the code should be closely related to the menu item, e.g, the keyboard entry might be composed of the first letter of the option label. See the guidelines in subparagraph 4.2.8.1.1.2.1 for the choice of abbreviated forms.

r. Arbitrary numbers or codes should not be used for keyed entry. Numbering options might be used when the list of items is particularly long, but this should be avoided (ref. 64).

4.2.8.1.2.2 INPUT AND OUTPUT

DEFINITIONS AND DESCRIPTIONS: There are the following two basic types of menu interaction: selection of one item out of several alternatives in order to perform a particular action, and selection of one or several items out of a list of alternatives in order to set parameters or select options.

The latter type of interaction involves the toggling of items between two states (e.g., bold on/bold off).

GUIDELINES:

a. Choices from menus should be made by a user in the minimum number of steps. For example, choices from a pull–down menu might be made by selecting the menu by positioning the cursor and performing the selection action, maintaining the selection action while moving the cursor down the menu (dragging), and ending the selection action when the cursor reaches the desired menu item.

b. Users should not be able to select menu items that are in conflict (e.g., two different font sizes in a text input task.) However, users should be able to select multiple menu items that are not in conflict (e.g., a font size and font type in text input). Each menu item selection would be a separate transaction with the system.

c. If the user is interacting with a hierarchy of menus, the user should be able to return to the initial menu with a single action.

d. Menus should not be used in cases where there is only one menu item. The use of menus for two menu items should be avoided in most cases. Methods other than menu selection for the input of one or two items are available (e.g., dialogue boxes and the UIL).

e. When menu items are not selectable, they should be identified as such to the user.

4.2.8.1.2.3 FEEDBACK

GUIDELINES:

a. When a menu item is selected, an immediate indication that the intended item was successfully selected should be given (e.g., making that item perceptually distinct). This indication should be consistent with menu selection indication in other menus and should not be confusable with other kinds of display codes and highlighting.

b. When a menu provides for the selection of several options or parameters, or when a menu item selection represents a continuing condition, some indication of selected items should be made (e.g., check mark next to item, reverse video, etc.).

c. In addition to the indication that the system has received the menu–based command, feedback about completion of the command should also be communicated. Completion of the action commanded by the menu item will be sufficient feedback, provided that the action has a result that is visible to the user. However, if the completion of the menu item has no visible result, the additional feedback that the command was completed should be communicated by a message in the Message Area.

d. For menu items that can be in an "On" or "Off" state, the "On" state should be indicated by making the item perceptually distinct.

e. Selection of menu items with "On" and "Off" states should change their state.

f. When a menu item is chosen by a keyboard entry, there should be some acknowledgement from the system that the item has been chosen (e.g., by highlighting the menu item).

4.2.8.1.3 FORMS

DEFINITIONS AND DESCRIPTIONS: Guidelines for the display of data forms presented in subparagraph 4.1.1.2.2.2 are relevant to the use of data forms as a dialogue type. When used as a dialogue type, a data form could provide a structured format for either data input or command inputs. The following guidelines describe additional requirements not addressed in subparagraph 4.1.1.2.2.2.

4.2.8.1.3.1 STRUCTURE AND REPRESENTATION

The structure and representation of the data form are described in subparagraph 4.1.1.2.2.2, Data Forms.

4.2.8.1.3.2 INPUT AND OUTPUT

GUIDELINES:

a. Distinctly different actions should be used for movement from field to field within a data form and indicating that the input to the form is completed.

b. Movement forward from field to field should be accomplished by a single action, such as pressing the Tab Advance key or the Return key (ref. 64, section 1.4.15).

c. Moving back to a field should be accomplished by a single action that differs from the action that moves the user forward in the data form, such as, pressing the Backtab key, simultaneously pressing the Control and Tab keys, or simultaneously pressing the Shift and Tab keys. In addition, the user should be able to move directly to a field by moving the pointing cursor to the desired field and, then, selecting the field.

d. If the user attempts to move forward from the final data field within a form, the action should position the placeholder cursor in the first data field of that form. If the user attempts to move backward from the first data field, the action should position the placeholder cursor in the last data field of the form.

e. Fields that are irrelevant to a particular task should be automatically skipped over and filled with null values or N/A.

f. Until the user enters the data from the data form, the user should be able to change any item easily.

g. Data forms are primarily used for entering data, but could be used for completing a command. If a data form is used for completing a command, the system should complete the action called for in the command unless the action is destructive (see subparagraphs 4.2.6.2.4 and 4.2.6.3) or for some other reason cannot be performed. The system should provide a message in the Message Area describing the status of the action and rationale if the action is completed or not performed (see subparagraph 4.2.6.2.4).

h. The user should be required to make an explicit action to exit the data form and save, not save, or cancel the data.

4.2.8.1.3.3 FEEDBACK

GUIDELINES:

a. Numeric fields should not allow the entry of inappropriate characters. An attempt to enter an inappropriate character should result in an auditory signal and an error message in the message area. The action that removes the error message should be entry of data into the data field, rather than explicit removal of the window containing the message.

RATIONALE: The purpose of the error message is to inform the user about the nature of the error. The most direct way for the user to acknowledge receipt of the message is to correct the error.

b. When a data form is used to communicate commands and the completion of the action commanded has a result that is visible to the user, feedback should be communicated by the completion of the commanded action. If the completion of the action has no visible result, feedback should be communicated by a message in the Message Area.

c. The system should acknowledge a command from a data form that cannot be completed by a message indicating non–completion of the command and an appropriate error message (see subparagraph 4.2.6.2).

d. The system should permit users to correct errors in command entries on data forms rather than simply rejecting the command. The command should be continued to be displayed so that the user can correct and reenter it. However, the user should have the ability to replace the command entry with any other command. (ref. 64, section 3.1.5.24.)

e. The system should acknowledge a potentially hazardous or destructive command as specified in subparagraph 4.2.6.3 (ref. 64, section 3.1.5.25).

4.2.8.1.4 QUESTION AND ANSWER

DEFINITIONS AND DESCRIPTIONS: The question and answer dialogue involves "a computer initiated sequence of transactions between the user and the system that provides explicit prompting in performing task and control activities" (ref. 3, p. 394). Question and answer dialogues are similar to forms in that the user provides input under the direct guidance of the system. However, question and answer dialogues are less constrained than forms, but generally require more time for input. Like forms, the question and answer would generally be used for inputting specific data but could also be used for inputting commands in a situation in which the command options were limited (e.g., Do you want to save or exit?). However, unlike forms, the question and answer dialogue is very amenable to branching; that is, asking the user questions based on his or her previous responses rather than a pre–established pattern.

4.2.8.1.4.1 STRUCTURE AND REPRESENTATION

GUIDELINES:

a. The system should provide the user with a specific request for information. The question structure should follow the guidelines for messages from the system to the user as described in subparagraph 4.2.6, User Guidance. A question mark should be the delimiter of the questions and answer dialogue.

b. In general, space for answering the question should be provided closely following the question mark. However, when additional information needed for the answer follows the question, the space for answering the question should be placed after the additional information.

c. The system should provide the user with contextual information required for answering the question. For example, if the only answer that the system would accept were a percentage, the question should be followed by "(%)". The answer area should follow the contextual information.

d. The system should accept as much information from the user as he or she provides in an answer. If the information that the system requests is constrained, a data form should be used.

e. The system should be able to stack questions and their associated answers if a series of questions were concerned with the same topic. The user should have the ability to remove a question and answer from the screen or recall a question and answer to the screen.

f. The system should display only a single question and answer if the following and preceding questions were not concerned with the same topic and should display only related questions simultaneously.

4.2.8.1.4.2 FEEDBACK

GUIDELINES:

a. When a question and answer dialogue is used to communicate commands and the completion of the action commanded has a result that is visible to the user, feedback should be communicated by the completion of the commanded action. If the completion of the action has no visible result, feedback should be communicated by a message in the Message Area.

b. The system should acknowledge a command from a question and answer dialogue that cannot be completed by a message indicating non–completion of the command and an appropriate error message (see subparagraph 4.2.6.2.4).

c. The system should permit users to correct errors in entries that answer a question rather than simply rejecting the entry. The entry should be continued to be displayed so that the user can correct and reenter it. However, the user should also have the ability to replace the entry. (ref. 64, section 3.1.5.24.)

d. The system should acknowledge a potentially hazardous or destructive command as specified in subparagraph 4.2.6.3 (ref. 64, section 3.1.5.25).

4.2.8.1.5 DIRECT MANIPULATION

DEFINITIONS AND DESCRIPTIONS: In a DIRECT MANIPULATION DIALOGUE, the user manipulates symbols in the display by directly interacting with the symbol. The SYMBOL can represent an object: such as, a data file; or an action: such as, run. The direct manipulation is generally performed through the use of a display structure: such as, a pointer; and a cursor control device: such as, a mouse. For example, a user might directly command that a data file be printed by moving the symbol for the data file to the area of the print symbol. For a fuller description of direct manipulation, see Hutchins, Hollan, and Norman (ref. 32).

4.2.8.1.5.1 STRUCTURE AND REPRESENTATION

DEFINITIONS AND DESCRIPTIONS: SELECTING is defined as the act of a user indicating that a screen element is to be readied for use. DRAGGING is defined as the act of moving a display element through parts of a display.

The symbols for objects are graphical representations, called icons, of the object. The symbols for actions are icons of the action or of an object used to perform the action. For example, the icon for the action, print, might be a line drawing of a printer.

GUIDELINES:

a. The system should use two principal direct manipulation actions: selecting and dragging. Selecting should involve the following two steps: indicating the object or action to be selected (e.g., moving a pointing cursor or other follower to an icon or function area) and indicating to the system that the icon and its associated function are required through the performance of a specific, well–defined selection action by the user (e.g., "clicking" a cursor control device button). Dragging should involve moving a selected icon or the cursor. Moving the cursor or dragging an icon can be accomplished by any of several cursor manipulation devices (see paragraph 4.3 for additional guidelines on moving the cursor).

b. The consequences of dragging should be contingent on the nature of the object that is dragged and where the object is placed at the termination of dragging. For example, dragging a data file icon to a "statistics" icon might cause the data to be analyzed; dragging the file icon to a disk icon might copy the file into that disk; dragging an icon to an unoccupied portion of the screen might simply move the icon and has no effect on the object.

c. Items on the screen that are selectable should be a minimum of 5 mm on a side and separated by at least 3 mm (ref. 74).

4.2.8.1.5.1.1 ICONS

DEFINITIONS AND DESCRIPTIONS: The guidelines that describe the visual features of icons are in subparagraph 4.1.1.1.6.4.

GUIDELINES:

a. The term used for the verbal label that accompanies an icon should be identical to the term for that function from the UIL.

b. The visual features, meanings, and specific uses of icons should be consistent within and between SSFP computer system applications.

RATIONALE: If users have to learn different associations between icons and the objects or actions that they represent for every different application or every different system, training times will be prohibitively long and errors are likely to increase.

c. Users should be able to move to and select icons, as well as move a selected icon, by use of any available cursor control device, including X–Y controllers and arrow keys (see paragraph 4.3 for additional guidelines concerning cursor control).

d. The user should be able to initiate the process related to a selected icon (e.g., opening a file or launching an application) in numerous ways. For example, the user might open a file by selecting a file icon and entering a UIL command, entering a keystroke command, choosing a menu item, or moving the file icon to an icon that represented the action, to open.

e. An icon that the user has selected should be highlighted.

f. Icons that are being moved should indicate where the icon originally was. For example, the solid line icon might remain at the original location while an outline of the icon followed under the cursor until the icon was no longer selected.

4.2.8.1.5.2 INPUT AND OUTPUT

GUIDELINES:

a. A user should be able to open an icon with a single unique action (e.g., pressing on a specific button of a cursor control device or a double click on the cursor control device button. Note: A "double click" is defined by two clicks within 700 ms of each other.).

b. In addition to using icons to represent objects and actions, other primary features of the direct manipulation dialogue should use windows for containing the data files (see subparagraph 4.2.3), and menus for additional objects and actions that are not easily represented by pictographic icons.

4.2.8.1.5.3 FEEDBACK

GUIDELINES:

a. Selection of an icon, menu, or application–specific capability from a function area should be acknowledged by highlighting the selected item.

b. If the completion of the action commanded by manipulation of an icon has a result that is visible to the user, feedback should be communicated by the completion of the commanded action. If the completion of the command has no visible result, feedback should be communicated by a message.

c. The system should acknowledge a command that cannot be completed by a message indicating non–completion of the command and an appropriate error message (see subparagraph 4.2.6.2.4).

d. The system should acknowledge a command not permitted by the system with a message indicating non–completion of the command and an appropriate error message (see subparagraph 4.2.6.2.4).

e. The system should acknowledge a potentially hazardous or destructive command as specified in subparagraph 4.2.6.3 (ref. 64, section 3.1.5.25).

f. During immediate execution mode, if the execution of a command statement will have adverse consequences, such as permanent loss of data, the UIL should request user confirmation prior to execution. The request should include a statement as to the exact nature of the consequences of executing the command statement.

4.2.8.1.6 NATURAL LANGUAGE

DEFINITIONS AND DESCRIPTIONS: In natural language dialogues the user and computer communicate in a conversational manner, with both the user and computer able to use and comprehend a flexible set of terms containing many synonyms and a flexible syntax. The only constraints to a natural language system would be the constraints of the language of the user, e.g., English. No natural language dialogue system has been developed and used successfully with computers.

GUIDELINES:

a. The system should not use natural language dialogues. However, the language used for commands, menus, forms, and question and answer dialogues should have as many features of English as is possible within the system and operational constraints.

RATIONALE: Natural language dialogues, as defined above, are not yet technically mature. Current technology would produce a substantial number of errors due to misinterpretations by the system and/or slow interactions as the system required additional explanatory or contextual information from the user. Accordingly, the safety and productivity of the Freedom Station would lead developers to reject the use of natural language at present. An additional concern about natural language is noteworthy, even if the technology becomes sufficiently mature to overcome concerns about safety and productivity. The Freedom Station computer systems will have many users for whom English is a second language and whose English language constructions may not appear to be completely "natural" to native English speakers. Such users may be better served by a UIL that has a stable and consistent syntax.

4.2.8.2 USER DEFINABLE DIALOGUE COMPONENTS

DEFINITIONS AND DESCRIPTIONS: User definable dialogue components allow users to assign a single component of the dialogue (e.g., a term in the UIL or a key in a set of function keys) to a single command or to a series of commands. The user can thereafter use that dialogue component to elicit those commands. User definable dialogue components include user definable macros (a single UIL term that elicits a command or a series of commands) and programmable function keys (a single key that elicits a command or a series of commands).

The HCI issues regarding user definable dialogue components, such as macros and programmable function keys, remain largely unexplored. One of the critical issues concerns the trade–off between the convenience and the user interface customization provided by a macro or programmable function key and the disruption of commonality introduced by macros and programmable function keys.

Additional advantages of macros and programmable function keys are that they do the following:

— Allow the grouping of related or sequential entries into one operation which may reduce errors such as forgetting a step or performing a step out of sequence.

— Provide for a rapid means of making a dialogue entry.

— Minimize the number of required keystrokes which reduces (macros) or eliminates (programmable function keys) syntax errors.

— Increase flexibility, for example, by allowing users to establish their own consistent terminology if they interact with different systems.

However, disadvantages of macros and programmable function keys are that they do the following:

— Reduce the commonality of the systems' interactive dialogues, which could lead to miscommunication between users and consequently, errors.

— Increase confusion; users may forget which macro names or programmable function keys they have specified.

— Place a greater burden on the user to remember the command(s) associated with the macro or function key.

— May increase errors and response time if users are inconsistent in specifying command names for the macros and/or if users assign more than one function to a programmable key and the assigned functions may be dissimilar (e.g., programmable function key three means save in one application and dump data in another application).

GUIDELINES:

a. Consider restricting the use of user definable macros and programmable function keys. The advantages may outweigh the disadvantages for some tasks (e.g., software development or modification) whereas, for other tasks (e.g., application specific software) the disadvantages may outweigh the advantages.

b. Consider providing a macro defining option which would impose syntactic constraints on a macro. For example, the macro defining option might provide a sequence of selections for naming the macro, for defining arguments, and for establishing the sequence of commands.

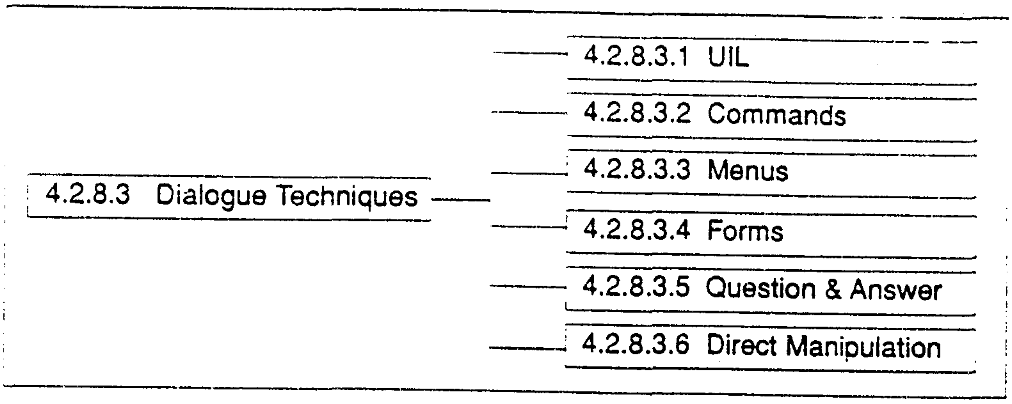
c. A user should be restricted from modifying a macro or programmable function key as defined by a different originating user; failure to do so could result in increased security risks.

d. Users should not be allowed to duplicate macro names; an error message should be provided to the user when he or she attempts to have a macro with a previously–used name.

e. Users should have access to an index of their macros and programmable function keys with their respective composition of commands.

f. Users who have macros and/or programmable function keys should be provided with information that aids communication (e.g., a list of the macro names and functions that they could make available to other users with whom they will communicate).

4.2.8.3 WHEN TO USE DIALOGUE TECHNIQUES



4.2.8.3 Dialogue Techniques

4.2.8.3.1 UIL

4.2.8.3.2 Commands

4.2.8.3.3 Menus

4.2.8.3.4 Forms

4.2.8.3.5 Question & Answer

4.2.8.3.6 Direct Manipulation

OVERVIEW OF SUBPARAGRAPH 4.2.8.3

GUIDELINES:

a. The dialogue types should match the needs of the user and the characteristics of the task. The specific guidelines for matching the users and tasks with the dialogue types are listed below.

b. The SSFP computer systems should allow users to switch from one dialogue type to another within an application, subject to the restrictions on the uses of each dialogue type.

4.2.8.3.1 UIL

GUIDELINES:

a. The UIL is especially well suited for tasks with an elaborate interaction between the user and the system. For example, a task in which a user needed to specify several actions or action objects or needed to modify the actions or objects would be served best by a command language.

b. The UIL should be a primary interface for highly trained, frequent users of SSFP computer systems (ref. 64, section 3.1.5.2).

c. The UIL may also be used in place of, or as a supplement to, other user–system dialogues. Accordingly, all users in all applications should always have access to the UIL.

4.2.8.3.2 COMMAND KEYSTROKES AND FUNCTION KEYS

GUIDELINES:

a. Command keystrokes should be used as a replacement for command language in cases where speed in command inputs is important.

b. The primary users of command keystrokes are likely to be highly trained, frequent users of the SSFP computer systems.

c. Function keys should be used for tasks with only limited (e.g, 10–20) unique control entries or as an adjunct to other dialogue types for functions that occur frequently and that must be made quickly and with minimal syntax errors. (ref. 64, section 3.1.4.1.)

4.2.8.3.3 MENUS

GUIDELINES:

a. Menus should be used primarily for tasks with a limited number of alternative actions and action objects in which the actions and objects need not be modified for the successful interpretation of the command by the system. In addition, menus should be used only when the actions can be grouped into a limited number of categories and menu items (e.g., less than 10) or can be organized into a logical hierarchy that is not overly complex.

b. Menus should be used when the computer response is relatively fast. (ref. 64, section 3.1.3.1.)

c. The primary users of menus are likely to be users who have had limited training or experience with the SSFP computer systems (ref. 64, section 3.1.3.1.), or users who have constrained, regular tasks.

4.2.8.3.4 FORMS

GUIDELINES:

a. Forms should be used for data entry and computer command tasks which are moderately constrained and in which users require some information to make their input. An example of such a task might be entering data and performing routine statistical analyses.

b. Forms should be used for tasks in which the user must make several data or control entries in a single step (e.g., choosing multiple control parameters). (ref. 64, section 3.1.2.1.)

c. Forms should be used for tasks in which the computer response time is slow. (ref. 64, section 3.1.2.2.)

d. The primary users of forms are likely to be moderately trained SSFP computer system users.

4.2.8.3.5 QUESTION AND ANSWER

GUIDELINES:

a. Question and answer dialogues should be used for highly constrained tasks in which each step in the task sequence has few choices available. Question and answer dialogues should be limited to routine data entry tasks and entry of commands in conditions with no more than four options available.

b. The primary users of question and answer dialogues are likely to be infrequent users of the SSFP computer systems who have received minimal training with the systems.

4.2.8.3.6 DIRECT MANIPULATION

GUIDELINES:

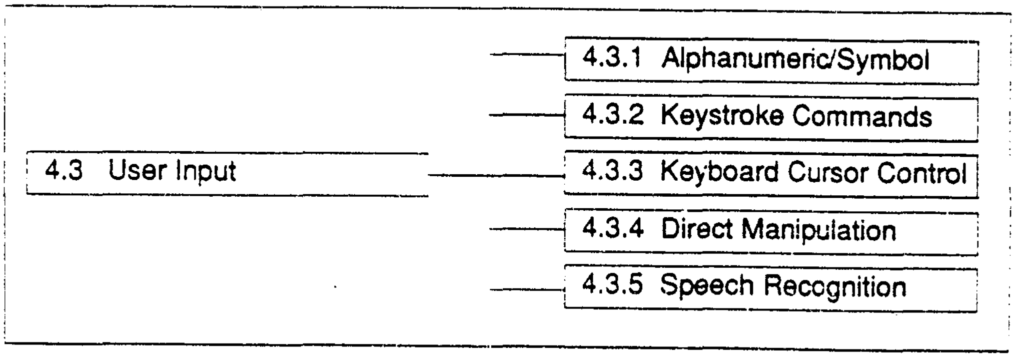
a. Icons should be used for tasks in which system users have different languages. (ref. 64, section 3.1.8.3.)

b. Direct manipulation should be used primarily in tasks with actions and objects that lend themselves to pictographic representation and in which the actions and objects need not be modified for the successful interpretation of the command by the system.

c. Direct manipulation should be used when the computer response is relatively fast. (ref. 64, section 3.1.3.1.)

d. Direct manipulation dialogues should be a primary interface for infrequent system users.

4.3 INPUT FROM THE USER


4.3 User Input
4.3.1 Alphanumeric/Symbol
4.3.2 Keystroke Commands
4.3.3 Keyboard Cursor Control
4.3.4 Direct Manipulation
4.3.5 Speech Recognition

OVERVIEW OF PARAGRAPH 4.3

DEFINITIONS AND DESCRIPTIONS: This paragraph of the Human–Computer Interface Guide (HCIG) focuses on the features of the SSFP computer systems' HCI related to the mechanisms by which users will input data, command the system, control the cursor, and manipulate data. The focus of this paragraph is on the guidelines related to input from the user that will affect software development. For the most part, the guidelines have been organized according to the type of input that the user makes (e.g., alphanumeric and symbol character input, X–Y control, X–Y–Z control, direct pointing, and speech).

See the current release of NASA–STD–3000, Vol. IV, for current SSFP requirements on user input hardware and the selection of user input devices for on orbit and MIL–STD–1472 (ref. 2) for ground–based systems.

GUIDELINES:

a. For free–drawn graphics, the refresh rate on the monitor should be high enough to produce the appearance of continuous track. (See the current release of NASA–STD–3000, Vol. IV, for current SSFP requirements on refresh rate.)

b. In a single monitor environment, movement of the controller shall be able to drive the follower only to the edge of the screen, not off the screen. (See the current release of NASA–STD–3000, Vol. IV, for current SSFP requirements on control devices.)

c. The consequences of any user input should be consistent from user to user and for any individual user across time. The consequences of the user's input should be closely linked to the input so that users can learn to predict what will happen any time they provide input to the system.

d. A user should be able to use any of a variety of input devices (e.g., a keyboard, a mouse, a trackball, etc.) to accomplish his or her tasks efficiently. However, a user's interactions with the Space Station Freedom Systems should be designed to minimize requirements for a user to alternate between input devices (e.g., between a mouse and keyboard or between a trackball and touch screen).

e. Control ratios and dynamic features of all input devices should permit the user to perform both rapid, gross positioning and smooth, precise, fine positioning (ref. 3).

f. Independent of control device and monitor type, movement across the display should be smooth and continuous.

g. Selectable items or regions should be, at a minimum, 5 mm on a side, but should not be so large that they waste screen space (ref. 74). In addition, larger selectable items or regions may not be perceived as selectable.

h. Selectable items should be separated by at least 3 mm. (ref. 74.)

i. When the user is required to return to the origin or other specific screen location following an entry or read–out (e.g., following an entry in a data form), automatic return of the cursor should be provided.

j. With multiple displays the location of the active cursor must be obvious to the user.

4.3.1 ALPHANUMERIC AND SYMBOL CHARACTER INPUT

DEFINITIONS AND DESCRIPTIONS: The KEYBOARD is typically used for a wide variety of tasks. It should be a primary control device for the input of alphanumeric characters, which should include the following functions: text input (including word processing and communications), command input (through use of alphanumeric characters in the command language and command keystrokes), numerical input, and software development. However, users may require other means of inputting alphanumeric and/or symbolic characters, such as selection of soft keys on a display via a cursor control device.

The alphanumeric and symbol character input keys do not include such keys as shift, caps lock, arrow, command, option, enter, etc.

GUIDELINES:

a. Alphanumeric and symbol character keys should automatically repeat when held down. The repeat should have a user selectable delay with a default of 0.5 second (see the current release of NASA–STD–3000, Vol. IV, for current SSFP requirements on keyboard input). In addition, the character should be repeated at a user selectable rate with a default of 0.1 second. The physical release of the key should terminate the repeat.

b. If the user redefines the keyboard (for example, by pressing a special purpose key to provide another set of characters, such as, Greek letters or scientific notation), a display of the new characters and their locations on the keyboard should be available to the user.

4.3.2 KEYSTROKE COMMANDS

DEFINITIONS AND DESCRIPTIONS: Keystroke commands may be input by simultaneously pressing a special purpose key and a specific alphanumeric key. In addition, function keys are typically used for the input of specific commands through a single keystroke, especially commands that occur frequently (e.g., enter, print, next page, previous page, options,etc.), are task critical (must be made quickly and without syntax errors), or are better made without moving the cursor (e.g., ditto, confirm, print, context sensitive help, and cancel) (ref. 64, sections 3.1.4.1–3). For additional guidelines on command keystrokes, see subparagraph 4.2.8.1.1.2.

4.3.2.1 FUNCTION KEYS

GUIDELINES:

a. Function keys should not repeat (except for delete, see 4.3.1 guidelines for repeating keys) (ref. 64, section 3.1.4.9).

b. Unneeded function keys, either fixed or programmable, should be disabled so that no other action occurs upon their depression except an advisory message. (ref. 64, section 3.1.4.1.2)

c. A legend of each key's function should be available to the user at any time upon request.

4.3.2.2 FIXED FUNCTION KEYS

DEFINITIONS AND DESCRIPTIONS: FIXED FUNCTION KEYS have a function that cannot be changed by the user or system and that remain constant between applications.

GUIDELINES:

a. The function assigned to a fixed function key should be standard across keyboards.

4.3.2.3 PROGRAMMABLE FUNCTION KEYS

DEFINITIONS AND DESCRIPTIONS: PROGRAMMABLE FUNCTION KEYS are user programmable and their function may vary between applications or between users within an application. A discussion of the advantages and disadvantages of programmable function keys is in subparagraph 4.2.8.2.

GUIDELINES:

a. Because programmable function keys are user dependent, a legend for each key's function should be available to the user at any time upon request.

4.3.2.4 KEYSTROKE COMMANDS BY SIMULTANEOUS KEY PRESSES

DEFINITIONS AND DESCRIPTIONS: Input of keystroke commands by simultaneous presses of two keys can be helpful to users who do not have access to a keyboard with function keys.

GUIDELINES:

a. A specially designated key (e.g., a Control key) should be one of the keys used for keystroke commands.

b. Keystroke commands should require the user to press both keys simultaneously, not in close temporal sequence.

RATIONALE: Requiring the user to press two keys simultaneously reduces the likelihood of inadvertent input of a command due to a missed keystroke that hits the specially designated key, followed immediately by another keystroke.

4.3.3 KEYBOARD–BASED CURSOR CONTROL

GUIDELINES:

a. Like alphanumeric keys, arrow keys should automatically repeat when held down. The repeat should have a user selectable delay with a default of 0.5 second (see the current release of NASA–STD–3000, Vol. IV, for current SSFP requirements on keyboard input). In addition, the movement increment should be repeated at a user selectable rate with a default of 0.1 second. The physical release of the key should terminate the repeat.

b. At the minimum, keys for cursor control should allow horizontal and vertical cursor movement. Ideally, keys for cursor control should allow horizontal and vertical movement, movement along the diagonals, and two or more rates of movement that are user selectable.

c. Users should be able to select at least two speeds (normal and fast) for the movement of the cursor when the keys for cursor control are held down.

d. Keys for cursor control should be oriented compatibly with the movement of the cursor that they produce.

4.3.4 DIRECT MANIPULATION CONTROLS

DEFINITIONS AND DESCRIPTIONS: Direct manipulation controls are defined by the close temporal and physical relations between the movement of the control device and the cursor, or other screen–based,follower (e.g., an icon or a window). Direct manipulation control devices include the mouse, the trackball, and pointing devices. In general, a direct manipulation device permits the user to move the cursor and to use the cursor to select a display structure (e.g., by clicking on a button on the device).

GUIDELINES:

a. The controller tracking speed (control display ratio) should be user selectable from a predefined list of alternatives (e.g., slow, 2:1; unenhanced, 1:1; moderate enhancement, 1:2; and high enhancement 1:3) but should have a moderate default speed.

b. If multiple clicks are required on the selection button, the user should be able to select the inter–click interval from a predefined list of alternatives. There should be a moderate default setting.

c. Rate aiding of the cursor movement (i.e., the speed of follower movement is proportional to the speed of input movement) shall be user selectable on/off. The default should be not to have rate aiding (zero–order control–display relation).

4.3.4.1 X–Y CONTROLLERS

DEFINITIONS AND DESCRIPTIONS: X–Y CONTROLLERS are generally used for the following: moving a cursor to and selecting display structures for subsequent manipulation, scrolling, data editing, and data retrieval. The specific devices that provide control over the cursor in the X and Y dimensions are the mouse, trackball, and the displacement and force joysticks. [The force joystick, also called an isometric joystick or a pressure joystick, is a lever that doesn't move, in contrast to the displacement joystick, also known as the isotonic joystick, a lever which the user can move (see the current release of NASA–STD–3000, Vol. IV, for current SSFP requirements on control devices)].

GUIDELINES:

a. The delay between the controller input and the resulting response on the screen should be less than 0.1 second. (See the current release of NASA–STD–3000, Vol. IV, for current SSFP requirements on control devices.)

b. The pointing cursor manipulated by an x–y controller should smoothly track the movement of the controller in the same direction, within +/–10 degrees without backlash, crosscoupling, or the need for multiple corrective movements. (See the current release of NASA–STD–3000, Vol. IV, for current SSFP requirements on control devices.)

c. In a single monitor/single controller environment, movement of the controller shall be able to drive the follower only to the edge of the screen, not off the screen. (ref. 3, p. 380, and see the current release of NASA–STD–3000, Vol. IV, for current SSFP requirements on control devices.)

d. The output of the force joystick should be proportional to and in the same direction as the user's perceived applied force. (See the current release of NASA–STD–3000, Vol. IV, for current SSFP requirements on control devices.) Similarly, the displacement joystick should provide output that is proportional to and in the same direction as the displacement of the joystick from the center.

e. Force feedback should be provided to the user of the force joystick, probably in the visual or auditory dimensions. (See the current release of NASA–STD–3000, Vol. IV, for current SSFP requirements on control devices.)

f. At a minimum, movement of the mouse across the entire maneuvering surface should move the cursor from one side of the screen to the other. Accordingly, control/display ratios should take into account both screen size and maneuvering surface size. (ref. 2, section 5.4.3.2.6.2.) This guideline should also apply to the displacement joystick moved to maximum displacement.

g. In a multitasking environment with multiple monitors, controllers, or cursors, careful consideration must be made so that the location of the active cursor is obvious to the user. If there are two pointing cursors, one on each of two monitors, the active cursor must be apparent to the user. If there is a single cursor that moves between two monitors, its path must be continuously trackable. As the cursor crosses from one monitor to the other, it should either maintain its vertical coordinate for side by side monitors and horizontal for stacked monitors or should jump between uniquely specified locations on each screen.

h. The controller should be able to produce any combination of x and y output values.

i. There should be minimal delay and tight coupling between control input and system response for both the displacement and force joysticks (ref. 3, p. 383).

4.3.4.2 X–Y–Z CONTROLLERS

DEFINITIONS AND DESCRIPTIONS: X–Y–Z controllers have the ability to control the cursor or other followers in the x– and y–dimensions and the screen and to provide control of apparent movement in the z–dimension. X–Y–Z control devices include the head movement controller, the glove controller, and six degree–of–freedom handcontrollers. Perhaps the best current applications for X–Y–Z controllers are for movement of three–dimensional graphics and proximity operations, although future applications might include movement through hypermedia databases.

A glove controller is a light weight glove–like device that transmits data records of arm, hand, finger shape, and position to a host computer. These data are provided by motion tracking sensors which transmit position and orientation of the hands as well as by flex sensing devices, usually located at finger joints, between fingers, and across the palm. The glove controller is useful for telerobotic manipulation or for manipulation of objects in a virtual environment display system.

HANDCONTROLLERS, as a general class, allow an operator to manipulate a small scale version of a larger and or stronger system. These include teleoperators or remote handling equipment. Several degrees of freedom may be exercised in positioning such a device including pitch, roll, and yaw of each of its sections.

GUIDELINES:

a. The delay between the controller input and the resulting response on the screen should be less than 0.1 second. (See the current release of NASA–STD–3000, Vol. IV, for current SSFP requirements on control devices.)

b. The pointing cursor or other follower manipulated by an x–y–z controller should smoothly track the movement of the controller in the same direction, within +/–10 degrees without backlash, crosscoupling, or the need for multiple corrective movements. (See the current release of NASA–STD–3000, Vol. IV, for current SSFP requirements on control devices.)

c. In a single monitor/single controller environment, movement of the controller shall be able to drive the follower only to the edge of the screen, not off the sceen. (ref. 3, p. 380, and see the current release of NASA–STD–3000, Vol. IV, for current SSFP requirements on control devices.)

d. The controller should be able to produce any combination of x, y, and z output values.

e. Movement of an X–Y–Z controller in the z–dimension should produce apparent movement of the follower in depth on the display (e.g., a decrease in size of a graphical object).

4.3.4.3 DIRECT POINTING CONTROLLERS

DEFINITIONS AND DESCRIPTIONS: DIRECT POINTING CONTROLLERS are used primarily for selecting display structures, controlling the cursor, and generating free–drawn graphics (ref. 3, p. 383). The control devices covered under these guidelines are the graphic tablet, touch pad, touch screen, and light pen.

A GRAPHIC TABLET consists of a stylus and a grid on which the stylus is moved. The grid may be either a transparent overlay on the screen or a remote pad.

The TOUCH PAD usually functions like a graphic tablet except that the user's finger replaces the stylus; however, some touch pads sense input relative to the cursor's position. For the latter case see X–Y Controller Guidelines.

The LIGHT PEN's control input is made relative to a grid that may be either a screen overlay or remote.

GUIDELINES:

a. Movement of the controller on the control surface should result in the smooth movement of the follower in the same direction at the same rate as that of the controller. Placing the controller at a point on the control surface should result in the follower appearing at the corresponding point on the screen. both when the control surface is the screen (or screen overlay) and when the control surface is remote; the follower should remain motionless until the controller is moved. (ref. 3, p. 384, and see the current release of NASA-STD-3000, Vol. IV, for current SSFP requirements on control devices.)

b. Placing the controller at the top of the control surface should result in the follower appearing at the top of the screen. This relationship should hold true even if the screen is in the vertical plane and the grid is in the horizontal (see the current release of NASA-STD-3000, Vol. IV, for current SSFP requirements on control devices).

c. For a touch screen, the minimum size of each selectable item or region shall be equivalent to the minimum size of a legend switch (15 mm on a side to allow for finger size and parallax inaccuracy) (see the current release of NASA-STD-3000, Vol. IV, for current SSFP requirements on control devices).

d. Selectable screen items or regions should be separated from each other by a sufficient distance to minimize inadvertent activation of adjacent items or regions. For example, the distance for key separation on a keyboard, 6.4 mm (see the current release of NASA-STD-3000, Vol. IV, for current SSFP requirements on control devices), may provide sufficient space between items that are selectable by pointing.

e. The delay between the controller input and the resulting response on the screen should be less than 0.1 second (see the current release of NASA-STD-3000, Vol. IV, for current SSFP requirements on control devices).

f. To avoid inadvertent activation of a selectable display structure, the system should accept only one controller input at a time and should recognize a controller input of approximately 0.1 second or greater for selection (ref. 3, p. 381).

g. Visual or auditory feedback (e.g., reverse video, color, or a tone) should be provided to indicate that a controller input has been registered. This is especially important when the control surface does not depress, thereby providing little tactual feedback to the user.

h. When using a touch screen, the cursor should be visible on the screen, offset from the point where the user's finger touches the screen and should be draggable as the user moves his or her finger. In addition, a manual action independent from cursor control (e.g., removing the finger from the screen) should be required for selecting an object or action.

RATIONALE: Marchionini and Schneiderman (ref. 46) have developed such a touchscreen strategy that has provided acceptable user input.

4.3.5 SPEECH RECOGNITION

DEFINITIONS AND DESCRIPTIONS: Speech recognition permits a user to provide spoken input which a computer interprets as data or commands.

Speech recognition is typically considered for the following applications and work conditions:

— Simple data–entry tasks (ref. 47, ref. 54, ref. 61, ref. 77);

— Data–entry activities with high task difficulty in a time–sharing situation (ref. 61);

— Verbal and selection tasks. However, complexity and time–sharing demands on cognitive resources must be taken into consideration.

— The user is a non–typist (ref. 17).

— The work environment is extremely dark (ref. 43).

— Neither panel space nor displays are available at the worksite (ref. 43).

— Hands are busy, tired, or gloved.

— User can not be at a stationary worksite while information is being input.

Speech recognition typically is not used in the following applications:

— A spatial task, i.e., cursor positioning or drawing (ref. 47, ref. 9, and ref. 49);

— An analog task, i.e., a character repeat or a drag task (ref. 47);

— The information to be input is of a private or classified nature (ref. 52).

GUIDELINES:

a. The SSFP user interface should not preclude the use of spoken input.

4.3.5.1 SYSTEM CHARACTERISTICS

DEFINITIONS AND DESCRIPTIONS: In recognizing a command, the system compares the input to every item in the vocabulary set. The system assigns a number representing its certainty that the input matches that vocabulary item. The system recognizes the item with the match of highest certainty. Rejection level is the minimum certainty (represented by a number) that this match must have for the command to be accepted.

The speech amplitude level is the required volume or sensitivity for speech input.

GUIDELINES:

a. Speech recognition systems should have an external, non–speech means of activation and deactivation (e.g., through a keyboard) so that extraneous conversation is not taken as command input (ref. 47). Additionally, if possible, a standby mode may be provided from which spoken commands to activate/deactivate may be invoked.

b. A consistent scale and/or the associated confidence rating which symbolize the similarity of each spoken command to the recorded template should be available the user.

c. Application vocabularies should be divided into sets based on the hierarchy of the application and recognition accuracy requirements.

RATIONALE: This improves recognition by reducing the number of choices that the recognizer has to consider to return the correct item.

d. The vocabulary items should consist of words that are meaningful and familiar to the user, be acoustically unique within a set, and consist of 2–5 syllables.

RATIONALE: Items of 2–5 syllables in length are generally better recognized than one–syllable items. (ref. 43, section 6.2.4.)

e. The speech amplitude and rejection levels required for input should be user–adjustable.

4.3.5.2 USING THE SPEECH RECOGNIZER

GUIDELINES:

a. The user should be able to test the recognition of any individual vocabulary item without the entire interactive system being on–line. Feedback on the word recognized and the corresponding confidence score shall be available immediately after each use of a word.

b. If an application functions with a speaker–dependent voice recognizer, the user should be able to retrain or update any or all vocabulary templates at any time.

RATIONALE: A user's voice changes over time, even in the course of an hour of continuous use (ref. 17). Several factors have the ability to alter the voice temporarily. To maintain good performance under these conditions, the user must have the ability to modify the template set.

c. Visual and auditory tone prompts, but not speech prompts, should be used during template training. (ref. 47.)

5.0 HUMAN FACTORS EVALUATION AND TESTING OF THE USER INTERFACE SOFTWARE

Paragraph 5.0 addresses the human factors evaluation of the user interface software as one part of the total Space Station Freedom Program (SSFP) computer systems. Although this paragraph discusses only the evaluation of the user interface software, several of the techniques described in paragraph 5.0 might be used during system evaluation. For an extensive discussion on conducting a human engineering evaluation of the user–system interface (including the interface with the hardware), see Military Handbook, Human Engineering Procedures Guide, DOD–HDBK–763, 27, February 1987. (ref. 4.)

The evaluation of an interface is essential to the successful design and operation of a system. Interface evaluation begins in the earliest stages of concept formation and design and continues through simulation and system development testing. No single evaluation procedure can adequately assess the user interfaces during each phase of development. However, as we describe here, a constellation of measurement techniques and evaluation methods can be used to cover the entire user interface evaluation process. Because the use of these techniques and measures varies depending on the level of interface development, the task, and the goal of the evaluation, this paragraph presents the methods in a format that outlines the advantages, disadvantages, and information provided by each. We also suggest the appropriate phases in user interface development in which to use each method. However, the ultimate choice of which evaluation technique to use and how to apply it rests with the evaluator.

Paragraph 5.1, Usability Engineering and Conducting the Evaluation: An Overview, presents six steps to follow in the evaluation process.

Paragraph 5.2, Evaluation Measures, presents several key aspects of an interface an evaluator should examine: errors, user response time, user preference, user judgment, training time, relearning time, and transfer of training. The definition of each aspect is contained in the appropriate subparagraph, as are typical methods of measuring each.

Paragraph 5.3, Evaluation Techniques, presents a set of techniques that can be used to evaluate an interface: controlled studies, questionnaires, protocol analysis (talking aloud method), debriefing, observational studies, Wizard of Oz method, and walk–throughs. This set is not an exhaustive list of all evaluation techniques but provides a representative sample. Each Evaluation Technique subparagraph describes the measures that can be used and the information each provides.

Paragraph 5.4, When to Use the Evaluation Techniques, presents two phases: design evaluation and system development testing, in their temporal order in the design–development process. Each subparagraph describes the evaluation techniques that can be used and the information each provides.

5.1 USABILITY ENGINEERING AND CONDUCTING THE EVALUATION: AN OVERVIEW

Interfaces are often designed and implemented with little consideration for the tasks that will be conducted on the system or the user who will conduct them. A thoughtful specification of the goals and functionality of a system increases the likelihood of developing a usable interface. One approach to interface design is usability engineering. Usability engineering is a process whereby the design objectives of an interface are specified quantitatively in terms of user performance objectives. These objectives guide the development of an interface and provide unambiguous evaluation criteria. Once designed, the usability of the system interface.is evaluated simply by comparing user performance against design objectives. As an aid, an outline of the usability engineering process is presented below. For a more detailed discussion of usability engineering, see Good, Spine, Whiteside, and George (ref. 27); Gould and Lewis (ref. 28); Bennett (ref. 10).

STEP 1: Understand the user and the tasks.

Information should be gathered which describes the users of the interface. A task analysis should be completed on the task(s) for which the interface is being developed. See Appendix C, Task Analysis Tables, for examples of task analyses.

STEP 2: Specify usability objectives.

Specific user performance objectives should be identified. These objectives should guide designers in developing the interface.

Objectives should be comprehensive, realistic, and quantifiable.

- -

EXAMPLE:

GOOD GOAL: Less than 3 percent error rate over one hour of performance for users with minimal (1–3 hours) training.

BAD GOAL: Users should perform accurately.

- -

The objectives should specify the aspects of the interface that will later be measured in an evaluation of the interface. It will be necessary to test more than one aspect of the interface. The interface which is most accurate will not necessarily be the fastest and the interface users most prefer may result in performance that is neither fast nor accurate. Because there will be tradeoffs among interface aspects, it is necessary to measure accurately as many aspects as scheduling and budget permit. Paragraph 5.2 provides detailed information on such measures as accuracy, preference, and training time.

Specifying usability objectives in a matrix may be useful, as recommended by Bennett, Butler, and Whiteside. The following is an example of a usability objectives matrix taken from Good, Spine, Whiteside, and George:

Evaluation Objectives	Technique	Worst Measure	Planned Case	Best Level	Current Case	Level
Initial performance task	Windowing benchmark	Work speed 1	Same as Version 1	20%> Version 1	3 times Version 1	Same Version
Initial evaluation	Attitude questionnaire score	Semantic differential	0	0.25	1	0.5

STEP 3: Implement human factors guidelines during design.

Apply the guidelines in paragraph 4.0 of this document in developing mock–ups, prototypes, and simulations.

STEP 4: Choice of evaluation technique.

The choice of evaluation technique will be guided by the usability objective, the aspects of the interface to be measured, and the stage of interface development (design stage, simulation stage, etc.). The techniques and measures, above all, must allow the evaluator to choose between alternative interfaces. An evaluation that ensures the success or failure of all users fails to provide the information that can distinguish good from poor interfaces.

STEP 5: Run the evaluation.

Collect data from people and cumulate the results. The people used as evaluation subjects should include the end user, Human–Computer Interface (HCI) design experts, software developers, and project managers. Specifically who is run in a given evaluation should relate to the objectives set in Step 2. A variety of statistical methods exist for cumulating data. The most appropriate statistics in many cases are the simplest: the mean, mean square error (variance), and minimum and maximum values. There are a variety of statistical analyses available to the data analyst, however, an extended discussion of statistical analyses is well outside the scope of this document. The interested reader may obtain information from texts by Keppel (ref. 35) and Bruning and Kintz (ref. 12).

STEP 6: Iterate until planned usability objectives are achieved.

Modify the design of the interface and re–evaluate until the planned usability objectives are achieved.

5.2 EVALUATION MEASURES

The ultimate objective of a well–designed user interface is to increase the productivity of the system user. The user's performance consists of a complex combination of response time or speed, accuracy of response, and preference or how well the user likes the system. The usefulness of the HCI can also be measured by its effects on training: the amount of time needed to learn to use the system initially, the amount of time to relearn to use the system after a long period of non–use, and the ability to transfer what is learned about one aspect of the system to another aspect. Consequently, no one evaluation measure will allow an adequate comparison of the costs and benefits of a set of interfaces. Choosing a method of calculating and combining evaluation measures thus becomes critical. Measures should be chosen, in part, on the basis of cost, schedule, criticality of the variable for system performance, and the evaluative appropriateness of the measure. In designing the evaluation, always keep in mind that proper system evaluation requires specific behaviorally–defined goals. Loosely defined goals or non–discriminating tests provide no basis for accepting or rejecting one interface over another.

5.2.1 MEASURES OF ACCURACY OF PERFORMANCE

Accuracy is the extent to which a task is completed without error. The concept includes errors of commission, omission, and deviation from standard rates or procedures.

5.2.1.1 PROCEDURAL ERRORS

For some tests accuracy may be defined as how far the operator progressed along a set of procedural steps. In these cases, performance would decrease when the user chose an incorrect or non–optimal step and went down a wrong path. This category of error is particularly frequent in hierarchical menu systems. Errors of this type increase the user's frustration with the system to the extent that the user has progressed far into the hierarchy before realizing the error or finds recovery difficult.

CALCULATING PROCEDURAL ERRORS

The following three measures of procedural error may be obtained: type, frequency, and depth. Error types are categories of the wrong responses made by the user. Examples include wrong command syntax, incorrect choice of a menu category, incorrectly selecting a menu item within the right category, and improper sequence of commands. Error types are normally defined by the task and usually involve progressing out of sequence. Error frequency refers simply to the number of deviations from the optimal path throughout a task. In contrast, error depth refers to the number of steps successfully completed or hierarchy levels successfully traversed before the user notices the error. In addition, these categories can be combined for a more specific description of the errors. For example, the relative frequencies of each type of error might show that the user interface was flawed only in specific areas that resulted in high frequencies of certain types of errors. Similarly, the depth of errors as a function of error type might show where in a sequence of steps an error was likely to occur.

Summary:

Error types: errors categorized by type

Error frequency: sum of deviations from the optimal path

Error depth: sum of steps taken away from the optimal path.

5.2.1.2 ERRORS OF CONFUSION

A metric closely related to procedural errors which may be appropriate for evaluation of the User Interface Language (UIL) is confusability. Confusability is the extent to which one word, function, or command is mistaken for another. For instance, an evaluator could design a study in which the actual commands entered by a user during a simulation or practice session are compared with the commands that should have been entered. Confusions are made apparent by using the data to construct a confusability matrix. Using the example given above, the actual commands would comprise one dimension of the matrix and the correct or optimal UIL commands would comprise the other. The commands are distinct if the diagonal of the matrix contains higher numbers than the off–diagonal. Confusion is high to the extent that off–diagonal values are large relative to diagonal values.

There is no metric for determining a minimally acceptable number of confusions among commands. A specific proportion of confusions out of the total UIL commands might be set early in the design of the UIL as the usability limit for a given command. Commands that produced a higher proportion of confusions (or conversely, a lower proportion of correct uses) would not be used in the UIL. In contrast, a developer may want to compare several different interfaces for confusability. With all else held constant, the one producing the fewest is preferable.

. . . .

EXAMPLE:

A very limited set of possible commands from a Power Management and Distribution subsystem is defined below:

Show – display a window of values or graphics related to a procedural step.

Set – constrain a parameter to a specified value

End – indicate the end of a procedural step

Close – constrain a toggled parameter to closed

Open – constrain a toggled parameter to open.

For purposes of the example, assume that a set of procedural steps by a user operating the subsystem was sampled from a larger number of procedural instructions. The procedures followed by the operator were nominal, and the set of commands issued to the system by the operator were captured and compared to what will be called the optimal commands (those that should have been issued according to the Flight Data File).

This comparison is displayed below:

STEP	OPTIMAL	ACTUAL
1	show	show
2	set	close;set
3	set	set
4	end	show;end
5	show	show
6	close	close
7	end	end
8	show	show
9	close	close
10	open	open
.	.	.
.	.	.
.	.	.

The first ten commands contain deviations from the optimal set shown by the multiple commands at Steps 2 and 4. These errors fall into the following two categories: using the wrong command to control a parameter and failing to end a procedural step before showing the values for the next one.

An example confusability matrix based on a large number of procedural steps for these commands might look like the following:

TABLE 5–1 ACTUAL COMMANDS

		ACTUAL COMMANDS				
		show	set	end	close	open
	show	31	0	1	0	3
OPTIMAL	set	0	25	0	8	0
COMMANDS	end	0	0	34	0	0
	close	11	0	0	21	0
	open	0	2	0	1	17

The confusability matrix would complement the results of the procedural error analysis. The easiest way to analyze the data in the matrix is to determine, for each specific command, the matches between actual and optimal commands (i.e., correct commands) as a proportion of the total commands. Thus, for "show" in this example, the user made 31 correct commands out of 35 total commands for a proportion of .89. In contrast, the user only made 21 correct commands out of 32 (.66) for "close," with all of the incorrect commands being "show." This suggests that the user confused show for close. (Note that the confusion between "show" and "close" was not symmetrical as evidenced by the high proportion correct when "show" was the optimal command.)

Commands found to be easily confused or procedures involved in many errors should prompt the consideration that aspects of the system require change. For this example, the low proportion of correct commands when "set" and "close" would be the optimal commands would indicate that these terms are easily confused with other terms in the command language.

5.2.1.3 ERRORS IN TRACKING

Many Freedom Station tasks will require users to monitor or control a system such that specified values or levels of instruments or parameters are maintained. For example, the scenario for Freedom Station reboost requires that an operator maintain the S.S. Freedom on a specific orbital path. Also, many subsystems will require that specific parameter values be maintained through continuous adjustment (maintaining a pressure level or flow rate, for example). For these tasks accuracy may be measured in terms of the frequency and extent to which the user–controlled parameters vary from specified standards. Whereas frequency counts of deviations provide some information, they do not indicate the severity of the deviation. Two common measures of variability that show the severity of deviation from the desired or expected value are the Mean Square Error (MSE) and the variance.

CALCULATING THE MEAN SQUARE ERROR

The MSE is the sum of the squared deviations of the user's performance from the standard (i.e., the optimal or required) performance divided by the total number of samples of performance collected. Represented mathematically:

$$MSE = \frac{\Sigma\,(u - s)^2}{n} \qquad variance = \frac{\Sigma\,(u - m)^2}{n}$$

where Σ = the sum of, u = user's performance, s = standard (optimal) performance, m = user's mean performance, and n = number of trials. The MSE should only be used when a performance standard can be determined before the evaluation.

The variance is very similar to the MSE. The sole difference is that the variance measures deviations from the user's mean performance, not from the standard. The decision to calculate the MSE or variance during an interface evaluation depends on how important the standard rate is to system operation. If there is a standard, it is usually best to calculate an MSE since the interface will be evaluated on the basis of how well the standard is maintained.

The number of trials is determined by the sampling rate and duration of the task. The numbe. of trials sampled should represent all aspects of the task. For example, suppose a system controller is required to decrease and stabiiize a pressure level. The most accurate estimates of the user's ability to perform this change are made when samples of the user's performance are collected at three points: before initiation of the pressure change and during the decrease and stabilization. The MSE or variance can be calculated at each of these three steps to determine the point at which the user encounters difficulty controlling the pressure.

INTERPRETING AND USING MEAN SQUARE ERRORS

Large MSEs or variances indicate large fluctuations in the value of the parameter being measured. Whether this is due to the system or the user may be assessed by testing numerous users and calculating MSEs or variances at different points during the data collection, such as the beginning, middle, or end of the task, or at critical points of the task. Testing numerous users avoids the danger of results based on idiosyncrasies. Collecting data at various points in the task may reveal differences in accuracy due to fatigue or confusing commands at critical task points. Using the pressure level example given above, the software evaluator can test whether greater errors occur during the pressure decrease or stabilization by comparing the MSEs from these phases of the task. Similarly, user vigilance can be examined by comparing errors occurring early versus late in the task.

EXAMPLE:

Assume that the operator of a Thermal Control System must, at one point, maintain accumulator pressure at 25 psi while performing other system–related tasks. The extent to which the interface and other tasks prevent the operater from maintaining the 25 psi standard can be evaluated by measuring the accumulator pressure and determining how closely the standard is met. The 25 psi level must be maintained for three minutes, and we can assume that measuring the psi level every 15 seconds provides a representative sample of the operator's performance. Assume also that a user was tested on two interface simulations, an analog display and a digital display. The results of this user, tested on each interface, show how the MSE can be derived.

The following two rows display the psi readings at 15 second intervals:

Analog dial: 23 25 25 26 28 26 25 24 24 25 26 23

Digital dial: 24 25 25 23 27 27 25 25 24 25 25 25

The mean reading from both the analog and digital displays is 25. Both displays were useful in that the average psi reading equals the standard value. In other words, the operator was able to maintain the required pressure levels with both interfaces, on average. However, an analysis of the deviations from the standard reveals that the two interfaces differed in the amount of variability around that average value. The calculation of the MSE for the analog dial is shown below:

u	s	(u–s)	$(u–s)^2$
23	25	–2	4
25	25	0	0
25	25	0	0
26	25	1	1
28	25	3	9
26	25	1	1
25	25	0	0
024	25	–1	1
24	25	–1	1
25	25	0	0
23	25	–2	4
26	25	1	1

The sum of the fourth column $[\Sigma(u–s)^2]$ equals 22. Since there are twelve samples of operator performance, n = 12 and the result

$$MSE = 22/12 = 1.83$$

is obtained.

The squared deviation values for the digital dial are shown below:

Digital dial: 1 0 0 4 4 4 0 0 1 0 0 0

The MSE for the digital display is 1.16. Whereas the use of the two displays resulted in similar average performance (the performance means are equal), the MSEs demonstrate that the digital display allowed the user to control the psi reading more consistently. Thus, the digital display would be preferred. Note that for this example the MSE and the variance are equal since the 25 psi standard equals the average psi value on each interface. In cases where users' performance with one interface met the standard but performance with another interface did not, the below standard interface would probably be judged to be inferior regardless of the relative MSEs.

Although this example shows a comparison between two interfaces, the MSE can be used as a measure in an evaluation of a single interface by following the approach outlined in paragraph 5.1. First, a limit for MSE should be set early in system design. Then, during the evaluation, the user interface would be tested to determine if it exceeded that limit. Redesign and retest would follow if the interface produced an MSE greater than the limit.

There are statistical tests that may be used to compare the means (or variances) calculated above, but a discussion of them is outside the scope of this document. The use of statistical tests requires a fair amount of knowledge and the interested reader may gain information about these techniques from a statistical text. Recommended texts are listed in paragraph 5.1.

5.2.1.4 ERRORS IN DETECTION AND MONITORING

Many aspects of subsystem monitoring and control require the detection of signals from the system in the form of instrument readings, Caution and Warning (C&W) messages, and other auditory and visual changes. One helpful measure of performance accuracy in these cases is the percentage of signals detected. However, the simple percentage of signals detected does not distinguish between the effect of the system on the user's response strategy and the user's ability to discriminate the signals generated by the system. For instance, many subsystems have a C&W system which a crewmember must monitor continually, at some level, but which rarely presents signals. Thus, the user does not watch the C&W display constantly; he or she carries out a number of tasks concurrent with monitoring the display. If a user operating a system fails to heed a warning signal, we cannot determine from a simple error count whether the user did not perceive the signal or the user was so busy with other tasks that the signal was misinterpreted or intentionally ignored. Accuracy measures specially designed for detection and monitoring tasks, called signal detection measures, may be more useful than simple percentage of detection because they separate the effect related to the signal from the effects related to the person doing the detecting.

The power of signal detection analysis can be demonstrated with a short example. A system operator may be assured of always avoiding a fully developed fault by acting as if the system is continuously approaching a fault. This strategy, however, would entail excessive cost and loss of productivity due to system inefficiency or downtime caused by frequent unnecessary fault management procedures. The opposite extreme is the operator who never responds to a warning signal. Although this operator will never cause unnecessary downtime, the disastrous consequences of this behavior are obvious. Nearly all operators fall between these extremes. Signal detection analysis separates the operators into those with liberal response strategies (those more likely to say a fault exists) and those with conservative response strategies (those less likely to identify a fault). More importantly, through signal detection analysis, the interface evaluator can separate the effects of the user's strategy from the effects of the interface itself, thereby determining the value of the interface.

Signal detection theory accounts for the range of possible user responses (hence strategies) to the range of possible system states in the following way:

TABLE 5–2 SIGNAL DETECTION RESPONSE MATRIX

		USER'S RESPONSE	
		YES	NO
SIGNAL	PRESENT	HIT	MISS
	ABSENT	FALSE ALARM	CORRECT REJECTION

The system has two states; it may present signals to the user (signal present) or not present them (signal absent). The operator can make one of two responses; he or she may either respond that a signal has occurred ("yes") or that it has not ("no"). Signals presented by the system and noticed or responded to positively by the user are termed "hits." Signals not responded to are "misses." A False Alarm (fa) occurs when the user indicates that a signal has occurred when in fact it has not. The final cell, Correct Rejections (cr), represents the condition in which the system presents no signal and the user correctly perceives this state. In the case of C&W, the signal would be the warning and the user's response would be some relevant action, e.g., to cease the current task.

Marginal probabilities within states of the system (the signal) sum to 1.0. That is, on any trial in which a signal is present the user will either score a hit (respond that the signal is present) or a miss (respond "absent"). Similarly, when the signal is absent the user will either score a correct rejection (notice it is absent) or a false alarm (mistakenly respond that the signal is present). Thus, P(hit) + P(miss) =1.0 and P(cr) + P(fa) =1.0, where P stands for "the probability of."

Knowledge of P(hit) and P(false alarm) permits a unique solution to the calculation of the user's response strategy (bias) and ability to discriminate signals (sensitivity). These two dependent measures are presented graphically in Figure 5–1, Signal and Noise Distributions and Criteria in Signal Detection Measures.

Noise trials (those in which the signal is absent) are represented in the distribution on the left in Figure 5–1. Signal trials are on the right. Most areas on the curves can be identified unambiguously; the areas that cannot are the overlapping tails of the distributions. Sensitivity identifies the extent to which the user can distinguish signal from noise. Sensitivity is related to relatively invariant aspects of the user's sensory system, thus it reflects the effect of the interface (whether the interface emits signals strong enough to be noticed by the user). Bias is related to variant aspects of the user, specifically, the user's response strategy. Bias describes whether the user is conservative (has a bias of reporting signals only when they are very strong) or liberal (reports signals when they are weak).

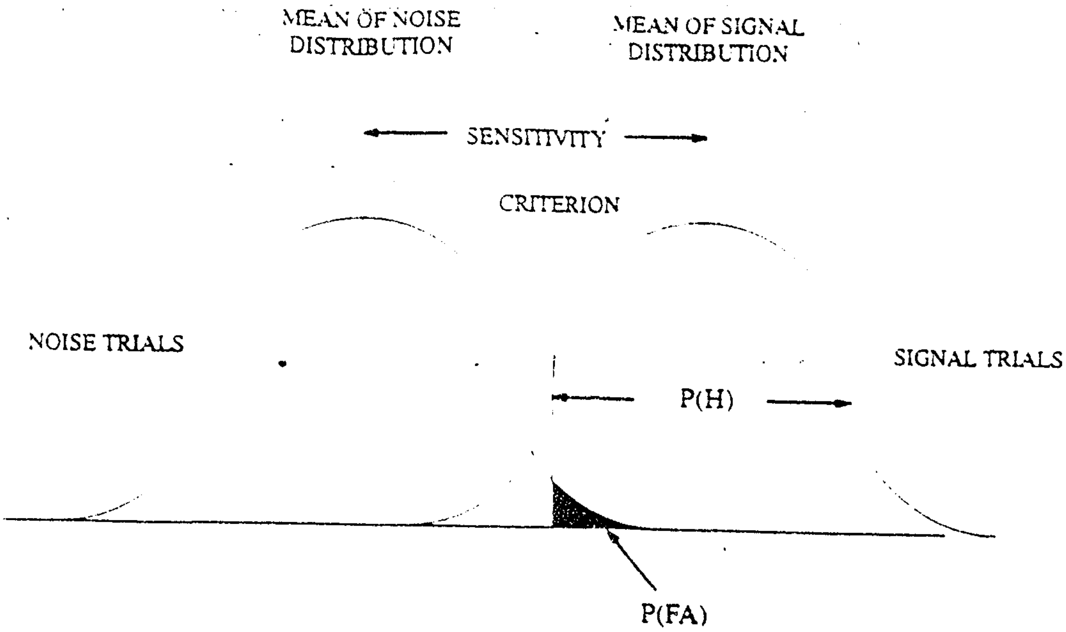


FIGURE 5–1 SIGNAL AND NOISE DISTRIBUTIONS AND CRITERIA IN SIGNAL DETECTION MEASURES

Consider the distribution of noise with no signal present. Noise is randomly distributed over time and, therefore, exhibits a Gaussian distribution (see the distribution labelled "Noise Trials" in Figure 5–1). The effect of adding a signal to the noise is simply to shift the noise distribution on the scale, leaving the distribution unaltered in any other way (see Figure 5–1 for a pictorial example). The task of the operator is to discriminate accurately between these two distributions. The "ideal" operator discriminates completely between a noise distribution and a signal + noise distribution; in other words, the two distributions are completely separated for the "ideal" operator. In contrast, for an operator who has no ability to discriminate between the two distributions, the signal + noise distribution would not be shifted from the noise distribution. Between these two extremes, signal detection theory allows an investigator to quantify an operator's sensitivity to the noise versus a signal.

CALCULATING SENSITIVITY AND BIAS

SENSITIVITY is formally defined as the number of normal standard deviations between the midpoints of the noise and signal distributions (Xs and Xn). This is the same value Xc would have if it were part of the noise distribution. The standard normal table (found in any statistics textbook) is used to find which portions of the two curves lie to the right of Xc. For P(h) = .85 and P(fa) = .10, the areas (z) equal 1.04 and 1.28, respectively. The two scores are added to compute sensitivity, thus sensitivity = 2.32.

BIAS is the ratio of the height of the signal to noise curves at Xc. From a table of ordinates of the normal curve, the height of the noise distribution at z = 1.28 is .176 and the height of the signal curve at z = 1.04 is .232. Thus, bias = .232/.176 = 1.32.

USING AND INTERPRETING SIGNAL DETECTION THEORY MEASURES

Signal detection measures are appropriate when the user's task requires substantial monitoring of displays. Because monitoring efficiency tends to decrease over time due to fatigue, measures of missed signals and false alarms are most informative in this situation. The taxonomic terms suggesting signal detection processes are "search," "scan," and "monitor." Because each of the four possible states of the system are represented in SDT it is necessary to choose tasks for which each signal has a response associated specifically with it, such that a clear option exists between responding and not responding. Other evaluation measures are more appropriate when tasks permit large delays in response time or have large components that do not require responses.

Greater sensitivity is represented by higher sensitivity scores. Bias values less than one indicate a bias toward reporting a signal (liberal response strategy). Bias values greater than one indicate a conservative response strategy. Because these measures are relative, testing more than one interface is necessary to determine which maximizes the user's sensitivity and sways bias in the desired direction.

5.2.2 RESPONSE TIME

Response time consists of two components, user response time and system response time. USER RESPONSE TIME is the speed with which a user can enter commands and control a system regardless of the computer's ability to quickly process the commands. SYSTEM RESPONSE TIME is the elapsed time between the initiation of a command and the notification to the user that the command has been completed. System response time guidelines are contained in paragraph 4.2.1 of this document and in NASA–STD–3000, Vol. 1, Section 9.6.2.d, and will not be addressed here.

For humans, as well as computers, speed of response is assumed to correlate with speed of processing. Items responded to quickly are often considered to require little mental processing; those associated with longer response times are considered to require relatively more mental processing. User response time may thus be taken to be a surrogate measure of the cognitive effort a person must expend before responding to a system query, prompt, or condition. Systems associated with short user response times are assumed to place fewer cognitive demands on users and are preferable to systems that elicit longer response times. In addition, shorter user response times lead to greater overall productivity, if accuracy is constant over different systems. Consequently, the interface that leads to the shortest response times would be preferred.

User response time may be defined in numerous ways; the simplest is the time elapsed between a request from the system for some action and the action. The request from the system may be either explicit (e.g., a prompt) or implicit (e.g., a display with information to which the user needs to respond to perform a task). As with all evaluation measures, user response time can best be used under specific circumstances. The user–computer interactions sampled for analysis of user response time should be those that place a premium on rapid response.

5.2.3 USER'S JUDGMENTS

Measures of user's judgments indicate the extent to which one system is preferred to another. User's judgments also provide a measure of what people think about elements of an interface (e.g., Does highlighting facilitate the task?).

Measures of preference or judgments are different from measures of performance. Assessing what users prefer about a system or what they think is a desirable design solution is a viable criterion. However, measures of performance suggest which design alternative should be chosen to reduce errors, increase user response time, etc. That is, performance measures assist in the selection of a design solution which enhances the efficiency of the system. Results from performance measures may be contrary to preference measures. For example, a user may prefer the use of a certain display color when performance data indicate that another color (or a monochrome display) results in better performance. Results from performance evaluations should generally be favored over results from preference evaluations for critical systems.

5.2.3.1 USER PREFERENCE MEASURES

Preference measures are used to assess the aspects of the user interface that the user finds helpful or engaging. To ensure relevant and accurate data, these measurements should occur immediately after the user's interaction with the system. However, "off the shelf" user preference measures do not exist. The advantage of preference measures is the ability to customize them to provide specific information. Toward that end, paragraph 5.3.2 provides a definition, advantages, and disadvantages, and examples for each measure. As with the other measures described, the preference measures chosen to evaluate the system should be based on a priori criteria and goals established in the initial design of the evaluation.

5.2.3.2 QUALITATIVE JUDGMENTS

Accuracy measures provide an assessment of differences in performance between two or more interface alternatives. Measures of qualitative judgments assess differences between interfaces based on what users perceive to be an effective or ineffective design. For example, users might respond to a questionnaire item such as "Does the title identify the display?" Like preference measures, measures of user's judgments can be constructed to provide specific information about an interface.

5.2.4 TRAINING–RELATED MEASURES

Training–related measures are used to assess the time required for a system user to achieve a specified level of competence in using the interfaces. The amount of time required to learn a system, relearn how to use the system, or the extent to which training on one interface transfers to another, all reflect how well the user interface is designed.

5.2.4.1 TRAINING TIME

Another measurable aspect of a system is the training time required for users to properly learn the commands and command sequences that control the system. If an operator (or two groups of equally competent operators) is tested on different interfaces, the difference in training time between the interfaces should directly reflect differences in the ease of learning the elements and operations of the interfaces.

Training time applies to the display and the UIL. Some measures of training time are as follows:

— The number of training classes or hours of classroom training needed to become proficient

— The number of practice sessions or hours of "hands–on" practice needed to become proficient

— The number of references to documentation during "hands–on" practice.

Examples of training time measures related to displays include the time necessary to learn the screen locations of information, the location of information within and across displays, or display order. The UIL and other dialogue techniques may be assessed by recording scores over time on tests of command definitions or actual command usage in experiments, prototypes, or simulations. Because of their closer fidelity to the task environment, tests of actual command usage and reference to documentation are probably more representative of user knowledge than verbal or written tests of definitions.

Note that many of the measures appropriate for accuracy, tracking, etc., are assessed with the addition of a time component when evaluating the training implications of an interface. For example, if users are equally accurate and fast with two different interfaces, the one which requires less training time is preferred.

5.2.4.2 RELEARNING TIME

If a delay occurs between the time a user is trained on a system and the time he or she uses the interface to control a system, then some amount of relearning will be necessary. RELEARNING TIME is a measure of the amount of work the user must accomplish in order to achieve a previous level of competence on the system. The training time measures discussed previously should be used to assess relearning. Relearning is likely to be affected by the training received by users on other systems or in other areas, particularly those users with many training courses and duties, such as astronauts and mission controllers. To evaluate the user interface, users tested for relearning time should be matched, to the greatest extent possible, on the basis of the amount of external training they have received.

5.2.4.3 TRANSFER OF TRAINING

Although a system should be consistent in its applications, the aspects of a system specific to any two applications will differ simply as a function of the different commands and actions required by dissimilar tasks. The features that optimize the operation of a robotic arm, for instance, will not necessarily be those that optimize search in a database. Nevertheless, an application interface should be sufficiently similar to the system interface to guide the user through the application with a minimum of guesswork and errors. The extent to which training on an application or system interface transfers to other applications may be assessed by training a user to proficiency on one application and testing his/her performance on a novel application. To the extent that the user trained on the first interface learns the second one more quickly/is more accurate/makes fewer procedural errors than a novice trained solely on the second interface, positive transfer of training has occurred. If training on the system interface or an application within the system resulted in longer training time or less proficient performance with a novel application, negative transfer of training has occurred.

5.3 EVALUATION TECHNIQUES

5.3.1 CONTROLLED STUDIES

A CONTROLLED STUDY is an investigation in which an independent variable (e.g., the type of display format or different dialogue techniques) is directly and systematically manipulated and the effects of manipulating that variable are measured. The purpose of the controlled study is to determine what relation exists between the independent variable(s) and the measures (dependent variables).

The six basic steps in designing a controlled experiment are as follows:

1. Formulate a hypothesis (e.g., UIL terms that resemble commonly–used English language terms will result in better performance and more rapid training than uncommon or idiosyncratic terms, e.g., UNIX terms).
2. Select appropriate independent variables and dependent variables (e.g., Independent variables — a set of common English terms, a set of uncommon English terms, and a set of idiosyncratic terms; Dependent variables — accuracy (or number of errors), response time, time to learn the meaning of the set of terms).
3. Control for other factors that may influence variations in the dependent variables (e.g., the groups of subjects trained on each set of terms should have equivalent experience with the English language and with the system).
4. Manipulate the independent variables and measure the dependent variables (e.g., train each group with one set of terms for a fixed period of time, then record the number of terms that they know, the accuracy, and the response time during a test session. An additional test that could be run would be to test the groups on a novel set of terms to assess transfer of training.).
5. Analyze the variance in the dependent variables (e.g., by use of Analysis of Variance or Multiple Regression).
6. Interpret the results (e.g., did the test performance of the group trained with common English terms exceed the test performance of the other groups?).

ADVANTAGES:

— Ability to control the application of the independent variable

— Ability to control other factors (e.g., subject participation, type of computer) that may influence the dependent variable

— Ability to record the dependent variable more precisely.

DISADVANTAGES:

— Controlled investigations can be more costly

— Controlled investigations may lack generalizability to other conditions (e.g., tasks).

5.3.2 QUESTIONNAIRES

Questionnaires consist of a series of items to which users may respond individually or in groups. Questionnaire items may include both close– and open–ended questions. Close–ended questions provide response options that can be worded to obtain specific detailed information. The response options may be a numbered scale, a choice from mutually exclusive items, or a percentage value (examples of each are given below). Open–ended questions allow the user to write freely in response to a question. The use of both types of questions ensures that the developer receives the specific information that he or she requires while the user can convey information about what he or she thinks is important or what the developer overlooked.

ADVANTAGES:

— Provide detailed information

— Are consistent across users

— May be customized easily

— Are easily scored.

DISADVANTAGES:

— May distort user answers by forcing them into the close–ended question format

— Extremely sensitive to poorly worded questions

— Low rate of return when not completed directly after testing

— Open–ended questions my result in irrelevant responses or a low proportion of usable data requiring extensive sifting through the data.

EXAMPLE:

Questionnaire items will vary according to the level of information desired by the evaluator and the particular interface evaluated. Example questions are given in Shneiderman (ref. 60, p. 400–407). This example presents some of the response scales that may be used to elicit user preference.

An example of a questionnaire item is:

I found the interface easy to use.

Users can respond to this question by circling a dichotomous choice:

Yes No

or by indicating their degree of agreement or disagreement with the statement:

1	2	3	4	5
strongly disagree	disagree	neutral	agree	strongly agree

An alternative to the dichotomous choice format is the bipolar adjective format:

Usability: easy difficult

When creating questionnaire items and choosing a response format, the evaluator must consider the time users have to answer questions, the likelihood that answers will accurately reflect the users' beliefs and knowledge, and the amount of information desired from the evaluation.

5.3.3 PROTOCOL ANALYSIS — THINKING ALOUD MEASURES

Protocols are verbal descriptions by the user of what he or she is doing, looking at, or attempting to do during task completion. These verbal descriptions provide insights into which aspects of the system are being used and the processes occurring during the task. One method for analyzing protocols is to sort statements into comprehensive mutually exclusive categories and calculate the frequency of statements in each category. For additional information on protocol analysis the reader is referred to Ericsson and Simon (ref. 22).

ADVANTAGES:

- Can provide insights that could not otherwise be collected
- May reveal user strategies
- May be videotaped and tape recorded unobtrusively
- May be less cumbersome than eye–trackers and other indirect measures of strategy or system use.

DISADVANTAGES:

- The data obtained will probably not be complete, possibly not even comprehensive
- Under many conditions, people have poor insight into their own strategies
- Technique depends heavily on the ease of describing the task and the verbal ability of the user.

5.3.4 INTERVIEWS

In an interview, the investigator engages a test participant in a discussion about the completion of a task. The discussion should be structured around a series of questions and/or topics to assure that the desired information is obtained in a minimal amount of time. Interviews should not be considered as a substitute for other evaluation techniques but should be used as one of several evaluation techniques. Data from an interview might be recorded by videotaping, audiorecording, or notetaking during the interview (either by the interviewer or by a person whose only job is to record the interview). The subject of the interview should be advised of any video or audio recording.

5.3.4.1 DEBRIEFING

Debriefing involves collecting information verbally from the user after task completion. The format of the debriefing may be described as an orally administered questionnaire: an interview based on a specific outline designed to address interface topics. The user is also allowed to address important topic areas and to ask questions for clarification or information. The evaluator is assured that vital or important areas are covered.

ADVANTAGES:

— Simple, low cost, and quick to implement

— Dialogue may provide information that would not have been extracted with the "monologue" format of the questionnaires, observations, or measures of accuracy.

DISADVANTAGES:

— Data collection by tape recorder in interview formats may be obtrusive

— Interviewer notes may be incomplete

— Interviewer and/or interviewee could introduce bias into the results

— Responses are highly dependent on the user's motivation, perception of the evaluation, and his or her memory of the events that occurred during the interaction with the computer.

5.3.5 OBSERVATIONAL STUDIES

Observational studies are sometimes referred to as natural experiments, because the study occurs in an environment the subject may experience everyday. The investigator or observer should be trained at identifying and recording behaviors (i.e., dependent variables) critical to the evaluation. Dependent variables that may be used in evaluating a user interface include task times and errors. In an observational study the investigator does not have control over the variables that may effect behavior. The variables that the experimenter lacks control over are the independent variables and other factors such as environmental variables (e.g., noise level in the room) and the selection of the people participating in the study. Otherwise, the approach to observational studies should be similar to that of controlled studies outlined below:

1. Formulate a hypothesis (e.g., users interacting with layered windows spend more time in window management functions, such as locating and moving windows, than do users interacting with tiled windows).
2. Select appropriate predictor (i.e., independent) variables and criterion (i.e., dependent) variables (e.g., Independent variable — type of window, layered or tiled; Dependent variable — amount of time in window management tasks).
3. Measure the dependent variables (e.g., By unobtrusively recording the amount of time users spend managing windows).
4. Analyze the variance in the dependent variables (e.g., by multiple regression).
5. Interpret the results (e.g., Did layered windows result in more window management activities than tiled windows?).

ADVANTAGES:

— May not be as expensive to conduct as a controlled study

— May allow for generalization to similar conditions

— Allows the investigator to obtain an overall impression of the subjects interaction with the interface.

DISADVANTAGES:

— Inability to control the application of the independent variable

— Inability to control other factors (e.g., subject participation, type of computer) that may influence the dependent variable

— Inability to record the criterion variable as precisely as in a controlled study.

Additional information about the design and use of observational studies can be found in Webb, Campbell, Schwartz, and Sechrest (ref. 76).

5.3.6 WIZARD OF OZ STUDIES

The Wizard of Oz method is based loosely on L. F. Baum's famous story (1900). An experimenter acts as the wizard to control what a user sees on a computer screen. The user is lead to believe he or she is interacting with an existing computer system (ref. 29). In implementing an Oz study, two machines must be linked to carry the communications between the user and the experimenter. The two machines can be either two computers (e.g., two IBM PCs) or a computer and a dumb terminal for the subject.

The Wizard of Oz method allows for acquiring a number of measures to evaluate the interface. Some of the measures to use in an Oz study may include errors of confusion, user response time, and preference measures.

ADVANTAGES:

— May not be as expensive to conduct as a controlled study

— Can provide insights that could not otherwise be collected

— May reveal user strategies

— May be videotaped and tape–recorded unobtrusively

— May be less cumbersome than eye–trackers and other indirect measures of strategy or system use.

DISADVANTAGES:

May lack generalizability to other conditions.

.7 WALK–THROUGHS

A walk–through is essentially a description to the intended user of how the user–interface operates by stepping him or her through the sequences of the interface. The display format and data entry requirements are described to the user in relationship to the user's task.

Information is collected verbally from the user during the walk–through. The investigator may have specific areas of concern for the user to answer. The user is also allowed to address important topic areas and to ask for clarification.

ADVANTAGES:

— Dialogue may provide information early in the design process with minimal cost.

DISADVANTAGES:

— Responses are highly dependent on the user's motivation and knowledge of the task.

5.4 WHEN TO USE THE EVALUATION TECHNIQUES

The three phases of evaluation technique use are discussed below: mock–ups, prototypes, and simulations. The phases form a continuum ranging from early in design and concept formation to full–scale system testing. The distinction between them has been blurred considerably by the advent of computerized rapid prototyping systems and object–oriented programming languages, but an attempt is made here to specify each as clearly as possible to guide design testing and system development throughout the product life cycle.

5.4.1 DESIGN EVALUATION

5.4.1.1 "PAPER" PROTOTYPING

"Paper" prototypes are early sketches of the information that is to be displayed in the user interface. They are called "paper" prototypes, even though they may be implemented either on paper or a computer. The term "paper" denotes the traditional method used to develop prototypes early in design, even though prototyping software currently available (e.g., Hypercard for the Macintosh) makes computer–based early prototyping more convenient than paper.

"Paper" prototypes are used very early in the design process, primarily to gauge the amount of information that may be presented on the display, to explore possible sequences of displays and combinations of information they may contain, to determine the compatibility of workstation and manipulation device configuration, and to evaluate the overall aesthetics and human compatibility of the system.

"Paper" prototypes are useful for early tests of user preference (the placement of information on the display, placement of manipulation devices, use of display area, display sequence, etc.) and accuracy and response time (finding elements on the display, grabbing or keying the cursor control device).

5.4.1.2 INTERACTIVE PROTOTYPING

Interactive prototypes are of higher fidelity than "paper" prototypes; they provide the user with the opportunity to interact with the user interface in basic ways, e.g., advancing display frames and manipulating keyboards, cursor control devices, and display elements. Interactive prototypes need not look like the final product but should in some form represent the functions and capabilities expected in the final system. Although interactive, a prototype will not necessarily control actual hardware or operate from a database. Prototypes are design evaluation devices; the functionality built into them should be only that required for an evaluation at an intermediate point in the design process, so that resources are spent creating numerous prototypes rather than one near–simulation.

Because they are more interactive than "paper" prototypes, interactive prototypes may be used to conduct more inclusive tests of response time (such as key presses) and accuracy (data input as well as extraction). Issues regarding training and learning time may also be examined with interactive prototypes. Users may also gain a better "feel" for the system; thus, user preference measures may be more accurate and stable.

5.4.2 SYSTEM DEVELOPMENT TESTING

Development testing is one of the final steps in the production of a system. Few designs should be under consideration at this point. One design for full–scale development should have been chosen using data from the "paper" and interactive prototypes.

5.4.2.1 SIMULATIONS

Simulations should represent the fullest task fidelity possible. This includes the display and keyboard formats, cursor control devices, workstation design, and peripheral surroundings. At a minimum, the simulation will be driven by a realistic database and provide real–time response. If possible, simulations should control end–product hardware or simulations thereof.

All evaluation measures may be employed in the evaluation of a simulation. A simulation provides the most realistic, stable, and accurate information regarding user performance and should be exploited to maximum benefit by t. system evaluator.

5.4.3 FINAL SYSTEM (OR PRODUCT) TESTING

In the final system test, a representative sample of system end users should perform actual tasks, interacting with the final software and hardware. At this point in the development process, major changes will be very expensive; accordingly, major problems with the user interface should have been identified early in the evaluation process so that users interact with the system rapidly, accurately, and with high preference levels in the final system test.

TABLE 5–3 EVALUATION TECHNIQUES AS A FUNCTION OF HUMAN–COMPUTER INTERFACE DESIGN STAGE

	System Design Evaluation		Development Testing
Evaluation Techniques	Mock–ups	Prototyping	Simulations
Controlled Studies	X	X	X
Questionnaires	X	X	X
Protocol Analysis		X	X
Interviews (Debriefing)	X	X	X
Observational Studies	X	X	X
Wizard of Oz	X	X	
Walkthroughs			X

6.0 EXAMPLE SPECIFICATIONS

6.1 INTRODUCTION

In general, Human–Computer Interface (HCI) guidelines documents have traditionally provided some pictorial examples but have not attempted to bring the guidelines together in a cohesive context–specific prototype. A realistic prototype provides pictorial examples of guideline ideas and, more importantly, illustrates the way in which the guidelines are implemented, the relationships between objects in the display, and the types of user interaction that are available. Prototypes are uniquely important to a guidelines document because they provide a means for integrating all of the ideas presented. Integrating the ideas contained in a set of guidelines is of special importance in the design of a system such as SSFP information systems that will have broad functionality and a heterogeneous user population. Paragraph 6.0 provides user interface prototypes based on the Human Computer Interface Guidelines (HCIGs) in paragraph 4.0 for the purpose of illustrating specifications and techniques for implementing the display and interaction guidelines presented in paragraphs 4.1 through 4.3 in the text.

CAUTION: The displays presented in this paragraph are not actual displays designed for production and future implementation, but serve as prototypes of possible HCI specifications and to demonstrate display techniques. In other words, although effort was taken to include accurate system information, the method of display and interaction is more important than the particular information shown. In addition, variations in tasks, hardware, and software, especially user interface management system software, will create different constraints for implementation of the guidelines than were placed on us in developing these prototypes. Accordingly, the displays presented in this paragraph include suggestions for implementing the guidelines but should not be read as excluding other, possibly equally effective techniques. Where possible, we have tried to use the most effective techniques available.

6.1.1 PROTOTYPING STRATEGY

The task analyses performed for representative SSFP information systems (see paragraph 3.0 and Appendix C) were used as the information base for prototype development. Variations in the type of tasks defined for each information system resulted in the development of prototypes that were stylistically different from one another. Some are primarily text oriented while others involve a more graphical interface. One of each type has been included in paragraph 6.0. All prototyping displayed and discussed here are based on working prototypes that were constructed using Hypercard on the Apple Macintosh $_{TM}$.

6.1.2 STANDARD SCREEN STRUCTURES

Commonality should be maintained among all displays comprising a single system. Users must be able to easily locate information of importance across a number of displays. Althougn the prototypes in this paragraph differ in style, care was taken to standardize certain generic screen structures. Figure 6–1 shows the basic screen structures that are.present on every screen. The top bar of the display area is a standard menu bar with two sections (see paragraph 4.1.1.1.6.3). The area to the left of the vertical line is reserved for pull–down menus available to any SSFP information system or application. The advantage of pull–down menus is that they require a minimum of display space and generally are displayed only when the user needs them, thereby reducing irrelevant information.

The "Resources" Menu in Figure 6–1 contains STP, NOTES, HELP, and SHOW BUTTONS. The selection of Short Term Plan (STP) results in access to the STP display seen in Figure 6–2; NOTES accesses a pop–over notepad as shown in Figure 6–3; and HELP accesses a pop–over help file as shown in Figure 6–4. The effect of selecting SHOW BUTTONS is that all "hot spots" or "buttons" (see glossary) on the screen are temporarily outlined. Pop–over or pull–down areas are advantageous because they do not permanently occupy display space, they are under the user's control, and they maintain the visual context because the background information stays on the screen.

The area to the right of the vertical bar is reserved for pull–down menus specific to the SSFP information system or application. Figure 6–1 shows the main menu for the Reboost operation; thus, the system specific menu "Reboost" contains items relating only to the Reboost system. To the right of these menus is an area for displaying mode information if applicable. Figure 6–1 shows that the mode for this Reboost example is "Simulation." The top right corner of the menu bar is reserved for an icon representing the system or application currently running. This serves as a constant visual reminder of the particular system that is active. It is also useful for the icon to be implemented as a "button" or "hot spot" that returns control to the main menu of the specific system.

The bottom bar in Figure 6–1 contains several standard areas. The far left rectangle contains a button labeled "Cmd Line." Activation of this button causes a window to be displayed for the purpose of entering the User Interface Language (UIL) (see paragraphs 4.1.1.1.6.2 and 4.2.8.1.1.1). The remaining areas contain Mission Elapsed Time, date, Greenwich Mean Time, and time at the Space Station Control Center (Central Standard or Daylight Time), respectively (see paragraph 4.1.1.1.6.6). Time is updated in the working prototypes at regular intervals through the computer system clock.

6.2 FREEDOM STATION REBOOST

6.2.1 REBOOST OVERVIEW

As stated previously, the requirements of a particular application will in part determine the style and components used to create the user interface. Freedom Station Reboost is an operation that will be performed at 73 day intervals; however, additional Reboost operations may be performed on a more frequent basis. Consequently, crewmembers are unlikely to remember specific commands and procedures for performing such an operation with accuracy. The infrequency of this operation and the magnitude of the consequences of a serious error suggest a user interface design that is very structured and that guides the user through each step of the procedure. Lack of familiarity with the procedure necessitates that the interface include textual explanations, error–checking and verification. A "Question and Answer" form of dialogue is most appropriate for this type of task (see paragraph 4.2.8.1.4). The interaction technique currently used in Reboost operations is a hard copy checklist. Because consistency with previous modes of operation is desirable, the user interface prototype for Freedom Station Reboost retains the checklist format with a "Question and Answer" form of dialogue. The tasks for the Reboost operation were taken directly from the Reboost Task Description (see Appendix C). Figures 6–5 through 6–21 show the series of displays presented in the Reboost prototype.

6.2.2 DETAILED DESCRIPTION OF THE REBOOST INTERFACE PROTOTYPE

After selecting BEGIN REBOOST from the "Reboost" menu, the user is presented with the display in Figure 6–5. The Reboost operation is presented in a tiled windows format. The tiled windows environment frees the user from the window maintenance required in an overlapping windows environment (see paragraph 4.2.3). The "Checklist" window in the upper left quadrant of the screen contains all of the steps in the procedure and is displayed during each step. The remaining space on the screen is occupied by windows that are needed for individual steps and are only present at the appropriate points. Each step in the checklist is associated with one of these special–purpose windows. The "Checklist" window scrolls automatically as each step is completed and verified by the system. The user initiates Freedom Station Reboost by using a pointing device (mouse) to "click on" or select the number "1." Figure 6–6 shows the result of this action.

The "System Status" window appears adjacent to the "Checklist" window. Note that the step indicator that is now pointing to number "1" in the "Checklist" window shown in Figure 6–6. This symbol always reflects the current step. The additional window that has appeared is necessary for completion of Step 1 in the checklist. One of the first tasks in the task description for Reboost (see Appendix C) is the evaluation of the condition of all SSFP information systems [i.e., Thermal Control System (TCS), Communications and Tracking (C&T), etc.]. Problems with any of the systems would indicate a possible delay of the Reboost operation. A response of "NO" would terminate the checklist procedure until all systems are nominal. Figure 6–6 shows all systems to be nominal, and thus the user will select the "YES" response in the "System Status" window.

Figure 6–7 contains the display for Step 2 in the Reboost checklist procedure. Note that the step indicator is now pointing to Step 2, and the checklist has scrolled such that the current step is at the top of the window. The user can click in the scroll bar area for the purpose of looking at past or future steps. Steps that have been completed are marked with an asterisk and are visually distinct from those steps not yet completed. In a checklist task, the user may need to see upcoming steps, as well as previous steps. Note also that the "System Status" window, since no longer needed, has been replaced by a "Prepare for Reboost" window. Notice the effort to limit the display of information to only that which is necessary for the particular step (as described in paragraph 4.1.1.2.1).

The "Prepare for Reboost" window requires the user to respond to each item before verifying that Reboost preparation has been completed. All items must be answered "Yes" before the user can advance to the next step.

Figures 6–8 through 6–12 show the displays that would be used to accomplish Steps 3 through 7 in the Reboost checklist. The user is required to review and verify information at each step and respond to a query before proceeding to future steps. This method of interaction guards against omission of steps or information oversights.

Figure 6–13 presents a graphical window of fuel and oxidizer temperatures. As this window is presented, the real time fluctuation of the temperature levels can be seen. Figure 6–14 shows the gauges at a specified level. As soon as the user verifies that temperatures are within the specified range by responding "Yes," the next window is presented. Note that the target levels for temperatures in Step 8 are displayed graphically. Also, notice the redundancy in the graphic and numeric representation of temperature values. For many decisions, the exact temperature values are not as important as the general range of temperatures; thus, the gauge is the better method of representation, especially when values are being constantly updated. The exact numeric value is also presented for the occasional instances that require this precision.

Figures 6–15 and 6–16 (Steps 9 and 10) present more information for verification. Figure 6–17 presents additional gauges to be checked by the user. Again, real–time values are reflected in the fluctuations of the gauge levels and the dial value. Figure 6–18 shows the instruments at specified levels.

Figures 6–19 through 6–21 show the displays for the remaining steps (12, 13, 14) in the Reboost checklist. The last display in the Reboost operation, shown in Figure 6–21 presents the user with an additional "Plot_Menu" which allows the selection of a number of different real–time plots. When "Earth" is selected from this menu, a full Earth view with orbital paths is shown. This allows the user to view the real–time effects of the Reboost burn on orbit. When the burn for Reboost has been completed successfully, the user may select any of the items in the "Resources" menu or select QUIT REBOOST in the "Reboost" menu. This last action would return the user to the main menu where selection of other SSFP information systems would be possible.

The Reboost operation serves as only one example of ways to implement the HCIGs. Please keep in mind that, as the Introduction stated, the limits placed on user interaction in this prototype would not necessarily apply to other applications. Especially, the infrequency of the Reboost operation and the seriousness of the consequences in light of an error mandate an interface that is tutorial and necessarily restrictive.

6.3 POWER MANAGEMENT AND DISTRIBUTION SYSTEM

6.3.1 POWER MANAGEMENT AND DISTRIBUTION OVERVIEW

The Freedom Station Power Management and Distribution (PMAD) System will be a large and diverse system requiring monitoring as well as interaction with functioning systems. Because of the wide range of PMAD functions, the human–computer interface for operating the PMAD is likely to be very different from the constrained checklist procedure used in Reboost. For example, the large number of instrument values and diagrams that the user must monitor in PMAD makes it well suited to an interface that is primarily graphical. The type of interaction that is most appropriate for a graphical interface is one that uses direct manipulation implemented through menus, buttons, and "hot spots"(see Appendix B, Glossary).

As described in the task analysis in Appendix C, PMAD consists of four subsystems: Power Management Control (PMC), Power Source Control (PSC), Main Bus Switch Control (MBSC), and Power Distribution Control (PDC). Figure 6–22 shows the main "PMAD" menu. Note that the general screen structures are the same as those in the Reboost prototype.

6.3.2 DETAILED DESCRIPTION OF THE POWER MANAGEMENT AND DISTRIBUTION INTERFACE PROTOTYPE

Each of the four Power Management And Distribution (PMAD) subsystems shown in Figure 6–22 can be accessed by "clicking" or selecting the appropriate rectangular button on the main PMAD display. These subsystems are also available on any PMAD display by pulling down the menu labeled "PMAD." The purpose for providing the user with access to the PMAD subsystems both through a pull–down menu and as buttons include the following: by presenting the user with all possible selections immediately upon entering the PMAD system in the buttons, the first display acts as an orienting/informational display; the action of "clicking" a button is very time–efficient; and the pull–down menu allows the user access to all other PMAD displays without using the limited display space.

Upon selecting one of the four subsystems, a very brief animation sequence occurs. The PMAD items shrink and move up into the "PMAD" menu and the Resource items shrink and move into the "Resources" menu. This serves as a quick visual reminder that the items will be available on all PMAD displays and will be accessible through the pull–down menus in the menu bar. The user is free to bypass this animation by directly selecting an item from the "PMAD" menu.

Figure 6–23 shows the first display for the PMC subsystem. The S.S. Freedom has three sources of power: Solar Array Sections (SAS), NiH Batteries, and Solar Dynamic (SD) mirrors. As the most common information queried in this system, the amount of energy available across all power systems is the only information shown on the initial display. If any systems (i.e., individual batteries) are not functioning, and therefore not contributing to the energy available, they are visually distinct (i.e., presented with dashed lines). In Figure 6–23 all power systems are functioning. This minimum information display allows a rapid query regarding amount of power available and production rates. Users can then request more detailed PMAD information or proceed to other system displays.

Figure 6–23 expresses the power level with a multidimensional symbol showing production rate along the "X" axis and capacity along the "Y" axis. A multidimensional symbol has the advantage of taking up very little display space while communicating a great deal of information. These kinds of displays are a good alternative to several tables of numbers or several graphical displays. Multidimensional symbols are used throughout the PMAD prototype. A user who needed information about the individual production systems could select the button labeled "Display Individual Systems" under the power–level display.

Figure 6–24 shows the result of selecting the "Display Individual Systems" button: a display that provides information about each of the power production systems. All are to some extent multidimensional symbols in that they each convey several pieces of information. The monitoring display shows the status of the SAS, the NiH storage batteries, and the SD mirrors. A system overview display should not be cluttered; therefore, Figure 6–24 only presents range information and system–flagged problems. The SAS Status display shows each numbered SAS and its temperature range. Note the temperature legend to the left. Additionally, out–of–range voltage and amperage values are flagged by the system and presented to the user as a "V" or "A" in the center of the SAS. The NiH battery display shows the temperature of the battery, its volume, and its charging status. The SD display shows the inlet and outlet temperatures of the SD fluid cycling through the receiver. These kinds of categorical range values are adequate for many monitoring situations and free the user from having to categorize precise values into critical ranges. A button labeled "More Detail" is available to allow users to view precise values. The visual effect of bringing in more detail is not that of viewing an entirely new screen, but rather that of the original display being altered by simply overlaying precise values. Figure 6–25 shows the detailed information display. A "Less Detail" button is available to return to the less detailed display. The detailed display is shown only by user request.

The PSC subsystem display shown in Figure 6–26 presents the following basic information that a PMAD System operator would use for decisions about System control: the orientation of the SASs and SDs relative to the sun in terms of alpha roll and beta angles, the current effect on power production, and the estimated time at this configuration to reach desired energy level. Users could choose default values for reorientation by selecting the appropriate box in the top left corner. A preview of the effects of this reorientation can be seen by selecting the box labeled "Show results of reconfiguration." Reorientation will not actually occur at this point; rather, selecting this preview merely displays a visual confirmation of the alpha and beta values. Alpha and beta values will change accordingly and the graphical representation of the arrays and mirrors will move to the commanded position. Once the user has determined that the arrays and mirrors are to be positioned as commanded, he or she could select the "Reorient" button in the bottom right corner. This choice would cause a window to pop up, thereby requiring the user to verify that the arrays and mirrors should be moved to the coordinates shown (see Figure 6–27). Upon verification, all structures would move to the appropriate positions, and the estimated time to achieve a specified level of energy would be recalculated and displayed. Rate of production, volume, and temperature are shown by the three dimensional storage cell symbol.

Figure 6–28 shows the MBSC display which consists of a diagram showing the paths, connection points, and switch gear locations for each of the main power buses throughout the Freedom Station. Voltage, amperage, and hertz values are shown on each bus line. The highlighted values are values that have been identified by the system as being out of range. These highlighted areas are also implemented as "touch zones" which allow the user to query the reason they have been flagged. When the pointing cursor is placed over a highlighted value and the mouse button is held down, a window temporarily pops up to explain why the value is out of range. As long as the mouse button is held down, the window remains in view; once the mouse button is released, the window disappears. For example, if a user "touches on" the out of range value "4A" shown in Figure 6–28, a small window appears under the highlighted area reading, "Amperage too low." "Touch zones" eliminate the window maintenance required when multiple windows are displayed on a screen; users do not have to take the time to clear away message windows. This technique also protects against screen clutter since the window is only present as long as the user is directly interacting with it. Although "touch zones" are very useful in this application, their usefulness is limited to very small messages because the user (as well as the system) might have difficulties if the mouse button had to remain down for extended reading.

The MBSC display also provides a button labeled "Show Default." Selecting this button causes a scrollable, pop–up window to appear in the center of the display (see Figure 6–29). This window contains a table of default voltage, amperage, and hertz values for comparison or reference. Once this window is not needed, a "click" anywhere inside the window hides it away. Figures 6–28 and 6–29 show ways to implement the ability to have large amounts of information available only when needed.

Figure 6–30 presents the display for the PDC subsystem. It consists of a basic diagram of the Freedom Station showing the percentage of power used in each area. Each area is implemented as a hot spot which the user could select to access a detailed display of that area in terms of layout and power requirements. Figures 6–31 to 6–39 show each of the detailed layouts that would be seen if the user selected that particular area of the power usage diagram in Figure 6–30. A user could return to the Power Distribution overview by clicking the arrow labeled "main diagram" which appears on every detailed screen. When the user has finished PMAD operations, QUIT PMAD could be selected from the PMAD pull–down menu appearing on every screen, returning the user to the Main Menu.

Unlike the Reboost prototype, PMAD utilizes a variety of graphical techniques for representing information. Some kinds of information are very easily accessible when presented graphically. In addition, PMAD is much less restrictive with regard to user interaction. For the most part, the user is allowed to choose the order of operations, bypass unwanted information or get more information whenever needed. The freedom allowed in this system would probably be inappropriate for an operation such as Reboost. Obviously, an understanding of the nature of the information to be presented is of utmost importance in selecting a user interface style.

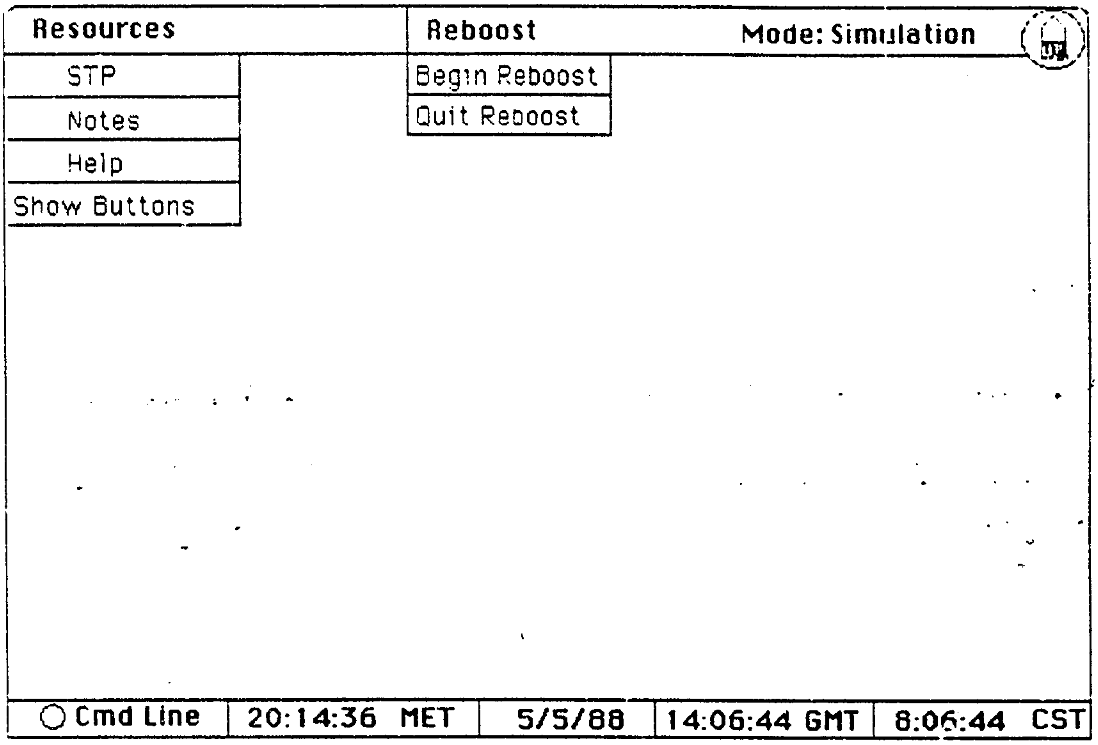
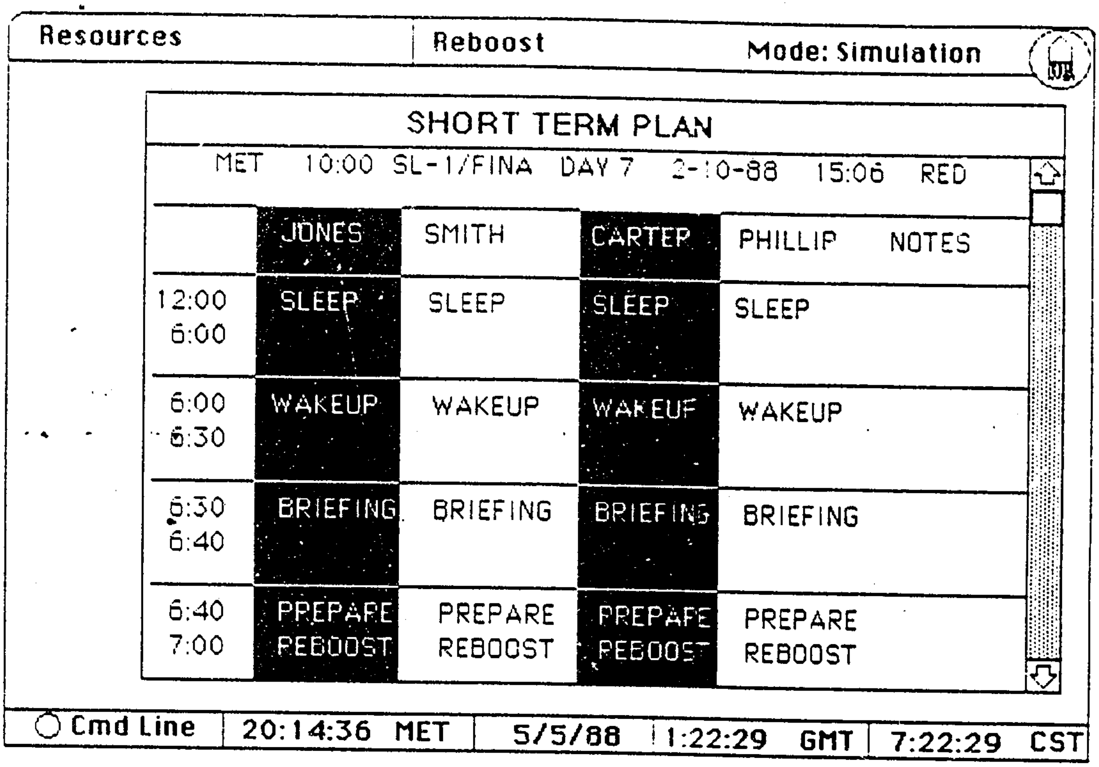


FIGURE 6-1 SCREEN STRUCTURES



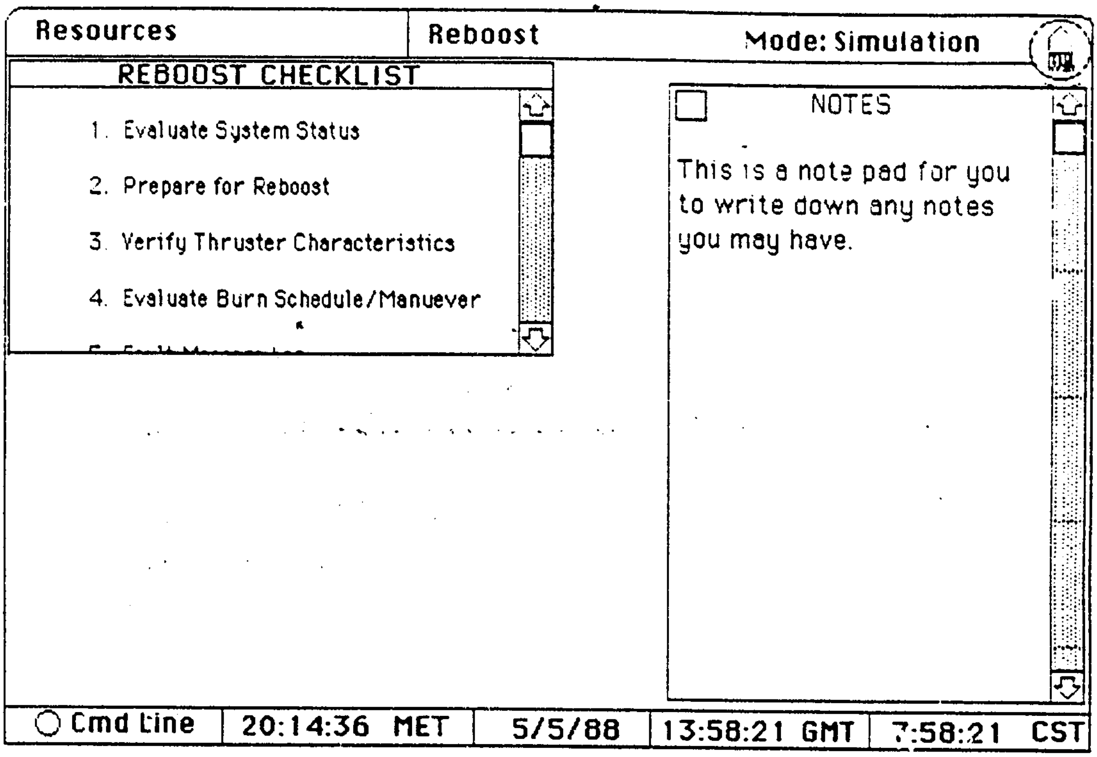
SHORT TERM PLAN

MET 10:00 SL-1/FINA DAY 7 2-10-88 15:06 RED

	JONES	SMITH	CARTER	PHILLIP	NOTES
12:00 6:00	SLEEP	SLEEP	SLEEP	SLEEP	
6:00 6:30	WAKEUP	WAKEUP	WAKEUP	WAKEUP	
6:30 6:40	BRIEFING	BRIEFING	BRIEFING	BRIEFING	
6:40 7:00	PREPARE REBOOST	PREPARE REBOOST	PREPARE REBOOST	PREPARE REBOOST	

○ Cmd Line | 20:14:36 MET | 5/5/88 | 1:22:29 GMT | 7:22:29 CST

FIGURE 6–2 SHORT–TERM PLAN



1. Evaluate System Status

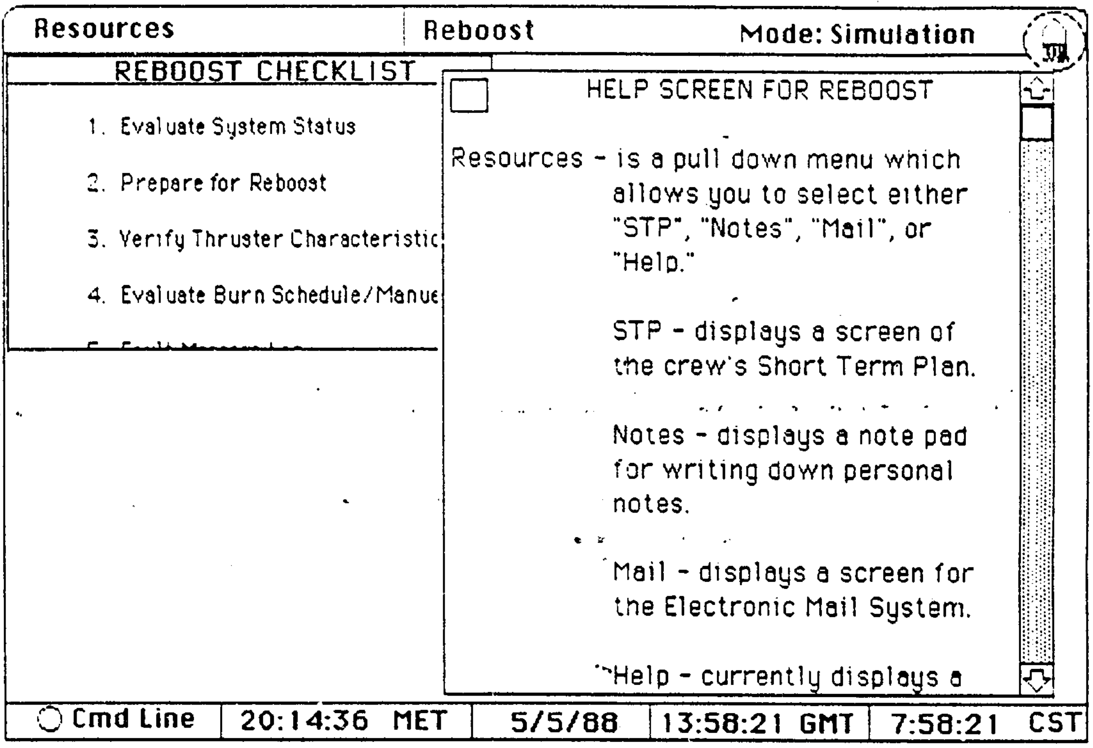
2. Prepare for Reboost

3. Verify Thruster Characteristics

4. Evaluate Burn Schedule/Manuever

This is a note pad for you to write down any notes you may have.

FIGURE 6–3 POP–OVER NOTES WINDOW



REBOOST CHECKLIST

1. Evaluate System Status

2. Prepare for Reboost

3. Verify Thruster Characteristic

4. Evaluate Burn Schedule/Manue

HELP SCREEN FOR REBOOST

Resources - is a pull down menu which allows you to select either "STP", "Notes", "Mail", or "Help."

STP - displays a screen of the crew's Short Term Plan.

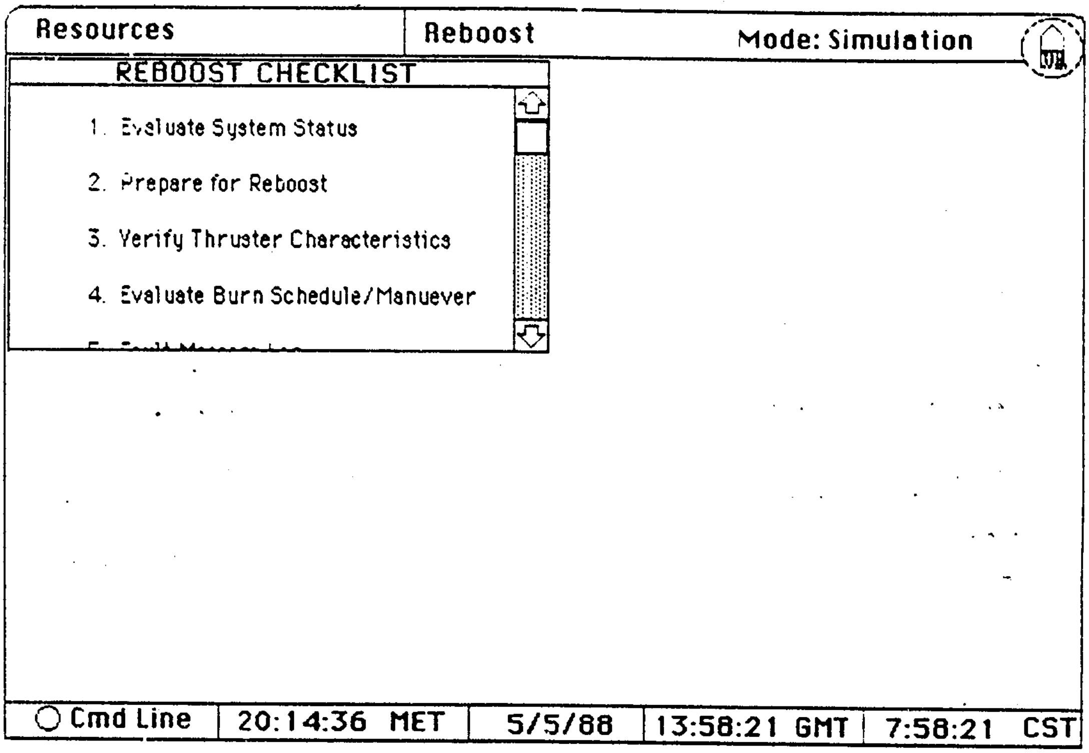
Notes - displays a note pad for writing down personal notes.

Mail - displays a screen for the Electronic Mail System.

Help - currently displays a

20:14:36 MET | 5/5/88 | 13:58:21 GMT | 7:58:21 CST

FIGURE 6–4 POP–OVER HELP WINDOW

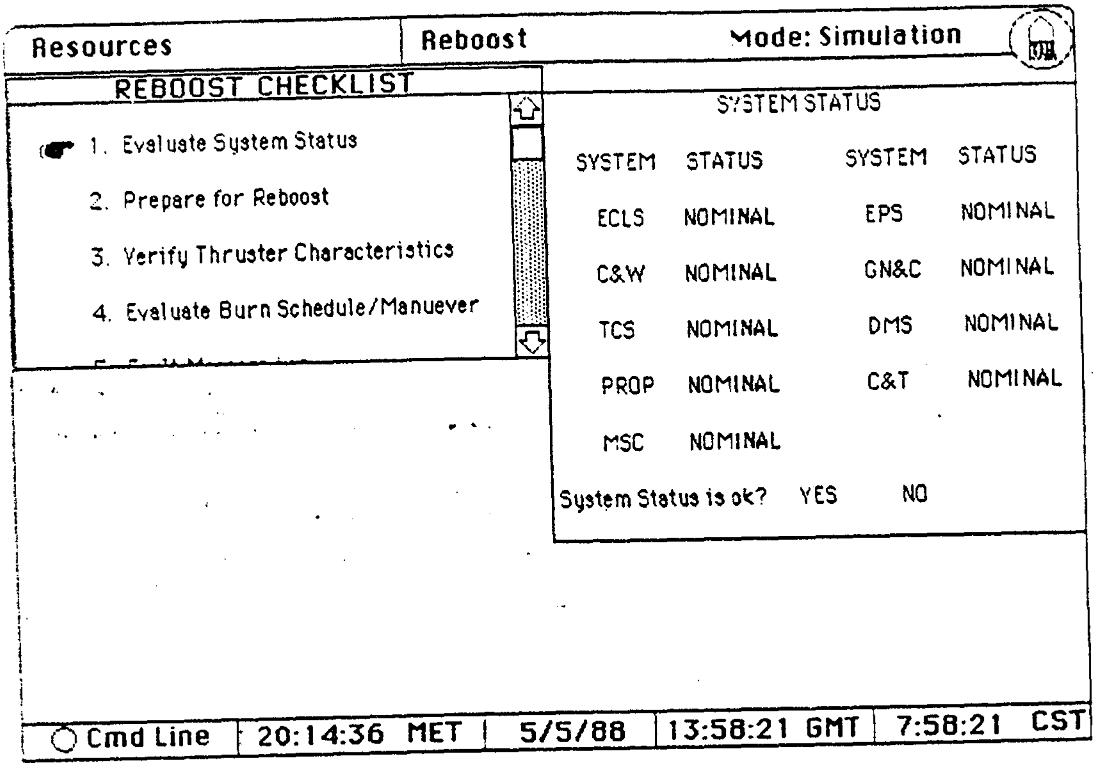


REBOOST CHECKLIST

1. Evaluate System Status
2. Prepare for Reboost
3. Verify Thruster Characteristics
4. Evaluate Burn Schedule/Manuever

○ Cmd Line | 20:14:36 MET | 5/5/88 | 13:58:21 GMT | 7:58:21 CST

FIGURE 6–5 INITIAL REBOOST DISPLAY



REBOOST CHECKLIST

1. Evaluate System Status
2. Prepare for Reboost
3. Verify Thruster Characteristics
4. Evaluate Burn Schedule/Manuever

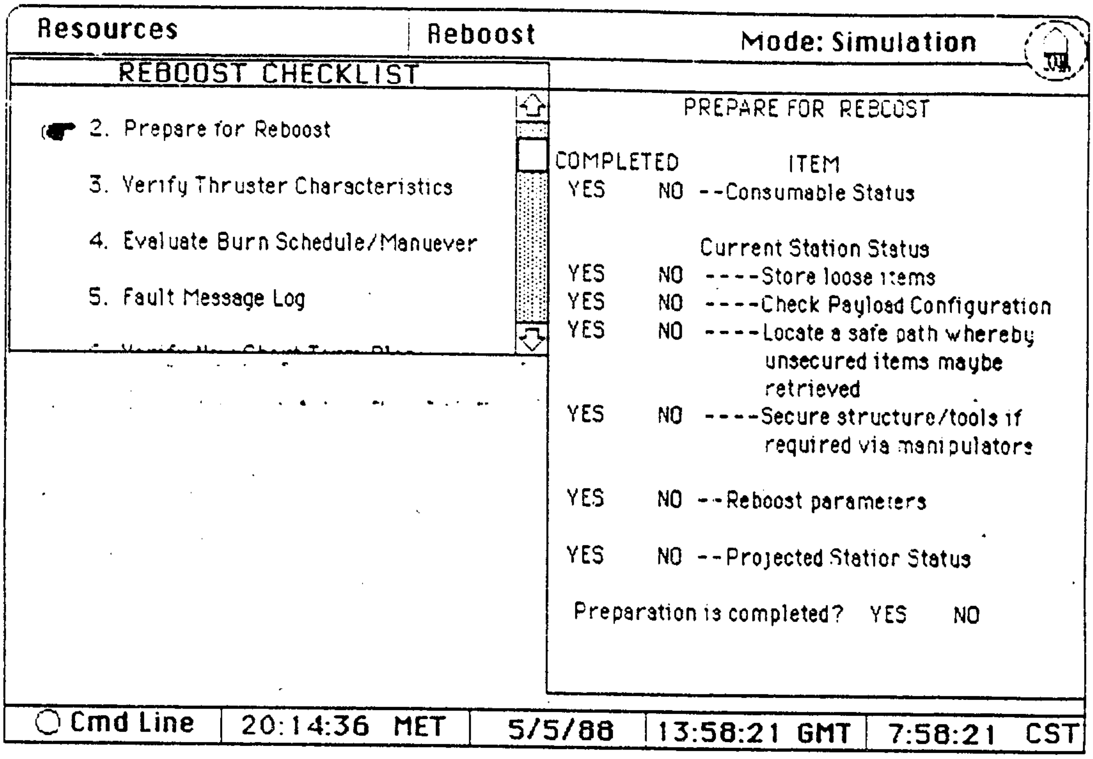
SYSTEM STATUS

SYSTEM	STATUS	SYSTEM	STATUS
ECLS	NOMINAL	EPS	NOMINAL
C&W	NOMINAL	GN&C	NOMINAL
TCS	NOMINAL	DMS	NOMINAL
PROP	NOMINAL	C&T	NOMINAL
MSC	NOMINAL		

System Status is ok? YES NO

○ Cmd Line | 20:14:36 MET | 5/5/88 | 13:58:21 GMT | 7:58:21 CST

FIGURE 6-6 REBOOST STEP 1



REBOOST CHECKLIST

2. Prepare for Reboost

3. Verify Thruster Characteristics

4. Evaluate Burn Schedule/Manuever

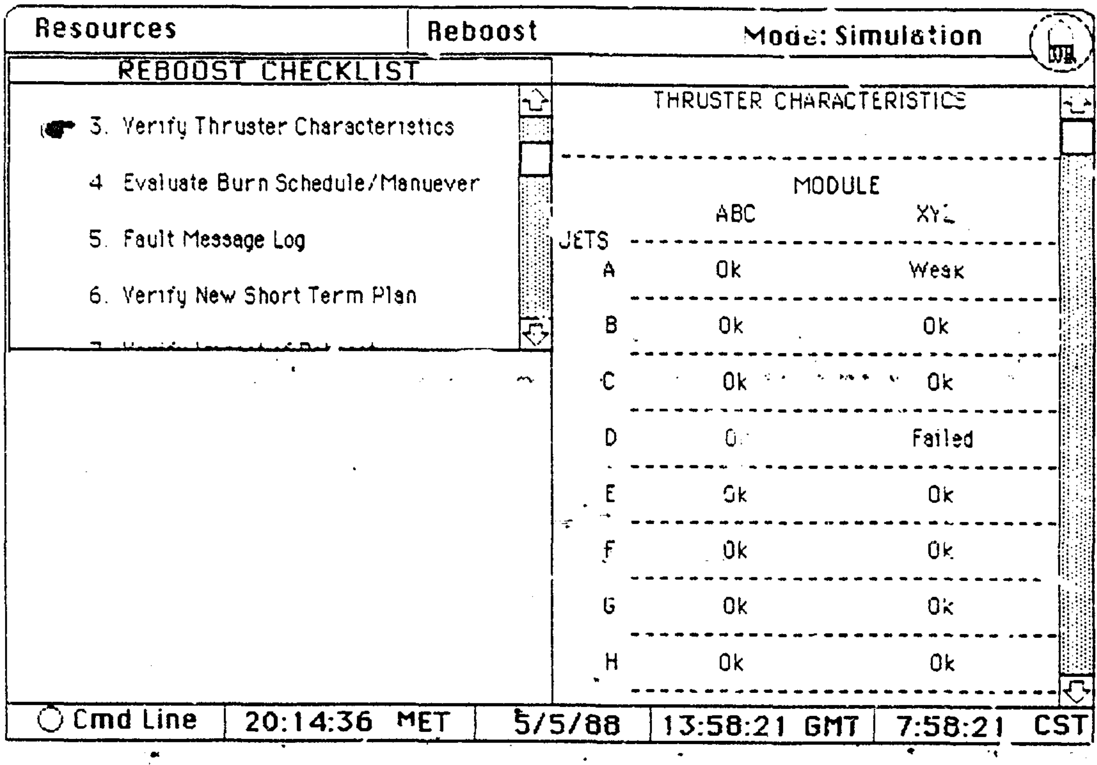
5. Fault Message Log

PREPARE FOR REBOOST

```
COMPLETED          ITEM
 YES      NO --Consumable Status

             Current Station Status
 YES      NO ----Store loose items
 YES      NO ----Check Payload Configuration
 YES      NO ----Locate a safe path whereby
                  unsecured items maybe
                  retrieved
 YES      NO ----Secure structure/tools if
                  required via manipulators

 YES      NO --Reboost parameters

 YES      NO --Projected Station Status

 Preparation is completed?   YES    NO
```

20:14:36 MET | 5/5/88 | 13:58:21 GMT | 7:58:21 CST

FIGURE 6–7 REBOOST STEP 2



Resources | Reboost | Mode: Simulation

REBOOST CHECKLIST

3. Verify Thruster Characteristics

4. Evaluate Burn Schedule/Manuever

5. Fault Message Log

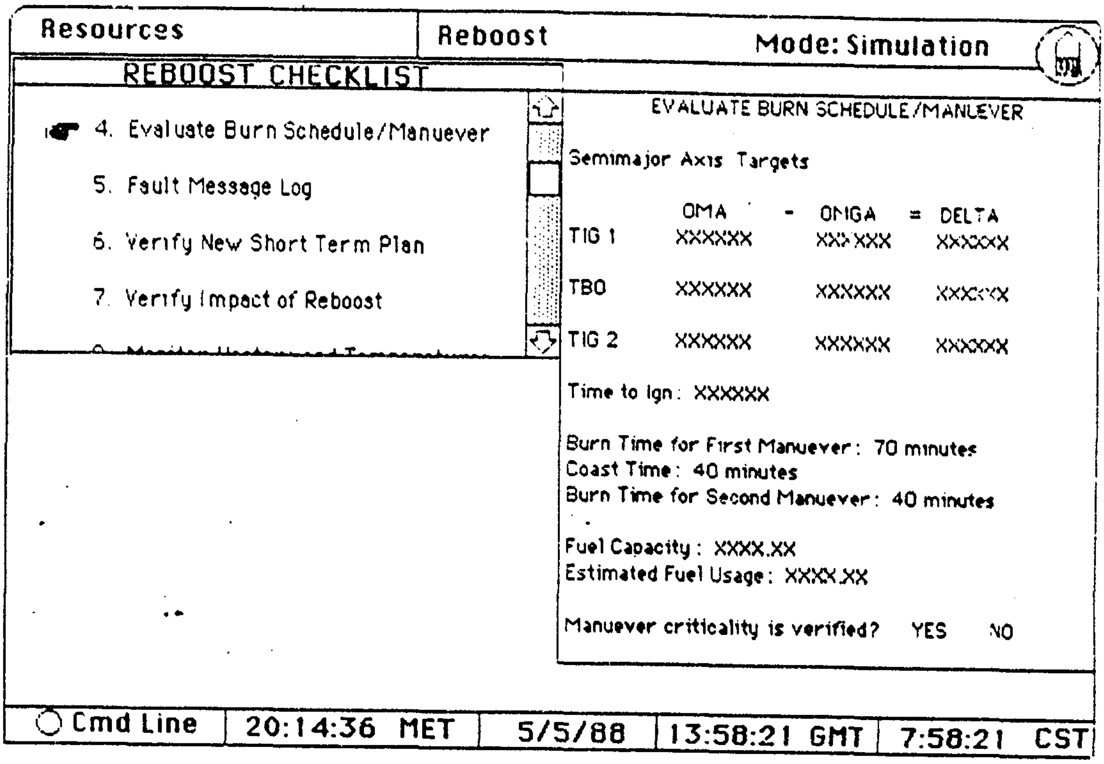
6. Verify New Short Term Plan

THRUSTER CHARACTERISTICS

	MODULE	
JETS	ABC	XYZ
A	Ok	Weak
B	Ok	Ok
C	Ok	Ok
D	Ok	Failed
E	Ok	Ok
F	Ok	Ok
G	Ok	Ok
H	Ok	Ok

○ Cmd Line | 20:14:36 MET | 5/5/88 | 13:58:21 GMT | 7:58:21 CST

FIGURE 6–8 REBOOST STEP 3



REBOOST CHECKLIST

4. Evaluate Burn Schedule/Manuever

5. Fault Message Log

6. Verify New Short Term Plan

7. Verify Impact of Reboost

EVALUATE BURN SCHEDULE/MANUEVER

Semimajor Axis Targets

	OMA	-	ONGA	=	DELTA
TIG 1	XXXXXX		XXXXXX		XXXXXX
TBO	XXXXXX		XXXXXX		XXXXXX
TIG 2	XXXXXX		XXXXXX		XXXXXX

Time to Ign: XXXXXX

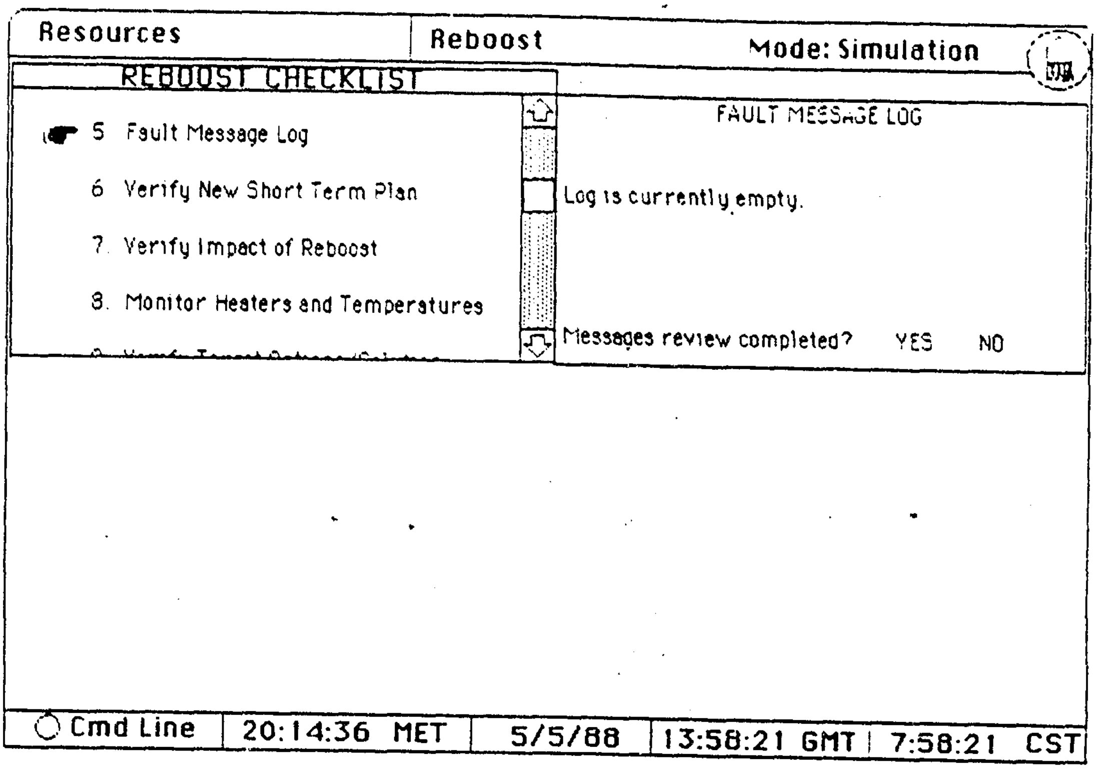
Burn Time for First Manuever: 70 minutes
Coast Time: 40 minutes
Burn Time for Second Manuever: 40 minutes

Fuel Capacity: XXXX.XX
Estimated Fuel Usage: XXXX.XX

Manuever criticality is verified? YES NO

○ Cmd Line | 20:14:36 MET | 5/5/88 | 13:58:21 GMT | 7:58:21 CST

FIGURE 6–9 REBOOST STEP 4



REBOOST CHECKLIST

5 Fault Message Log

6 Verify New Short Term Plan

7. Verify Impact of Reboost

8. Monitor Heaters and Temperatures

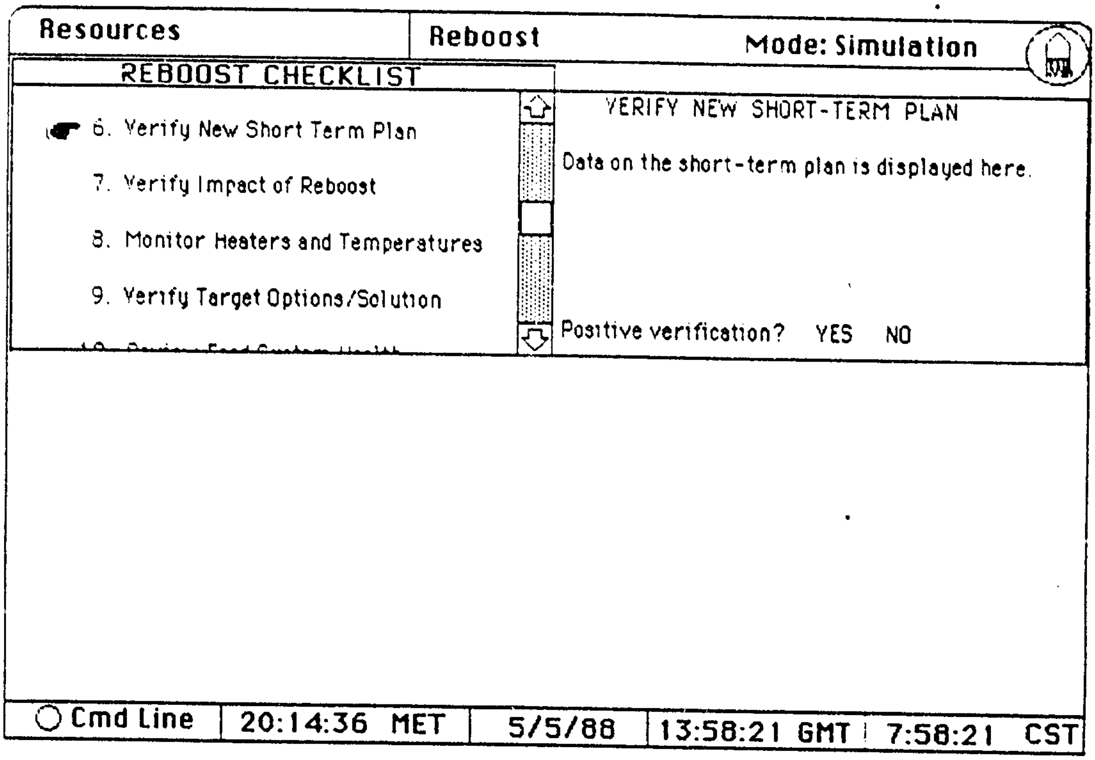
FAULT MESSAGE LOG

Log is currently empty.

Messages review completed? YES NO

○ Cmd Line | 20:14:36 MET | 5/5/88 | 13:58:21 GMT | 7:58:21 CST

FIGURE 6–10 REBOOST STEP 5



REBOOST CHECKLIST

6. Verify New Short Term Plan

7. Verify Impact of Reboost

8. Monitor Heaters and Temperatures

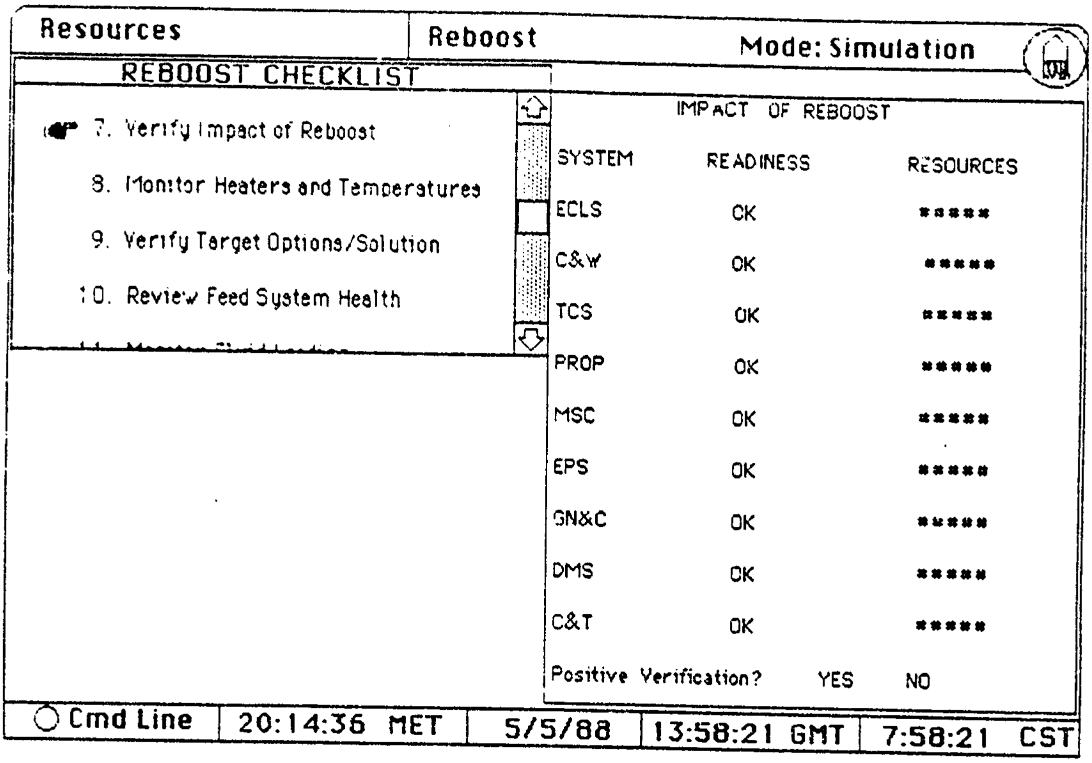
9. Verify Target Options/Solution

VERIFY NEW SHORT-TERM PLAN

Data on the short-term plan is displayed here.

Positive verification? YES NO

FIGURE 6–11 REBOOST STEP 5



REBOOST CHECKLIST

7. Verify Impact of Reboost

8. Monitor Heaters and Temperatures

9. Verify Target Options/Solution

10. Review Feed System Health

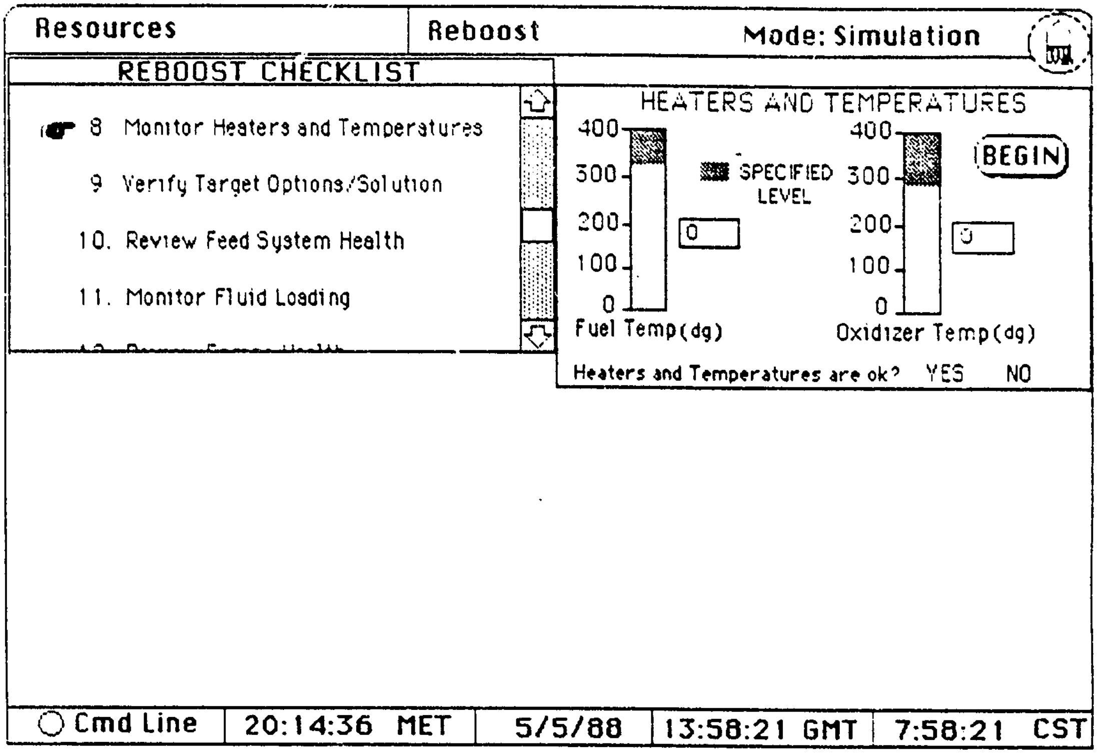
IMPACT OF REBOOST

SYSTEM	READINESS	RESOURCES
ECLS	OK	*****
C&W	OK	*****
TCS	OK	*****
PROP	OK	*****
MSC	OK	*****
EPS	OK	*****
GN&C	OK	*****
DMS	OK	*****
C&T	OK	*****

Positive Verification? YES NO

○ Cmd Line | 20:14:36 MET | 5/5/88 | 13:58:21 GMT | 7:58:21 CST

FIGURE 6-12 REBOOST STEP 7



8 Monitor Heaters and Temperatures

9 Verify Target Options/Solution

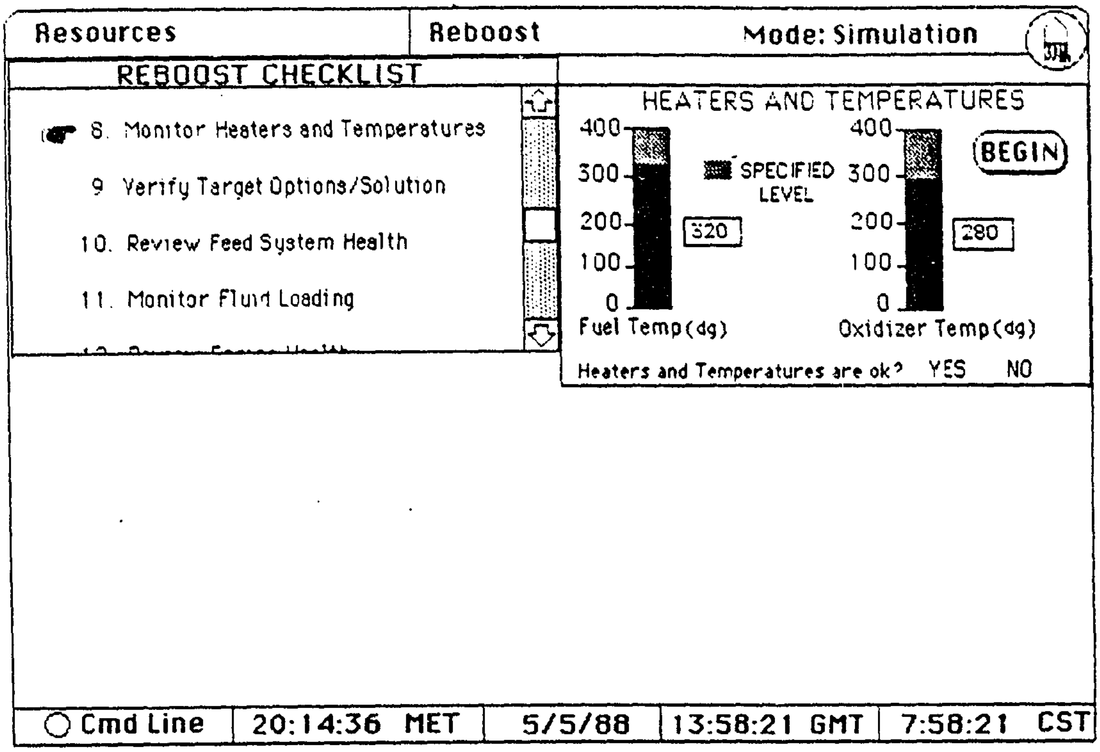
10. Review Feed System Health

11. Monitor Fluid Loading

Heaters and Temperatures are ok? YES NO

○ Cmd Line | 20:14:36 MET | 5/5/88 | 13:58:21 GMT | 7:58:21 CST

FIGURE 6–13 REBOOST STEP 8



Resources | Reboost | Mode: Simulation

REBOOST CHECKLIST

8. Monitor Heaters and Temperatures

9 Verify Target Options/Solution

10. Review Feed System Health

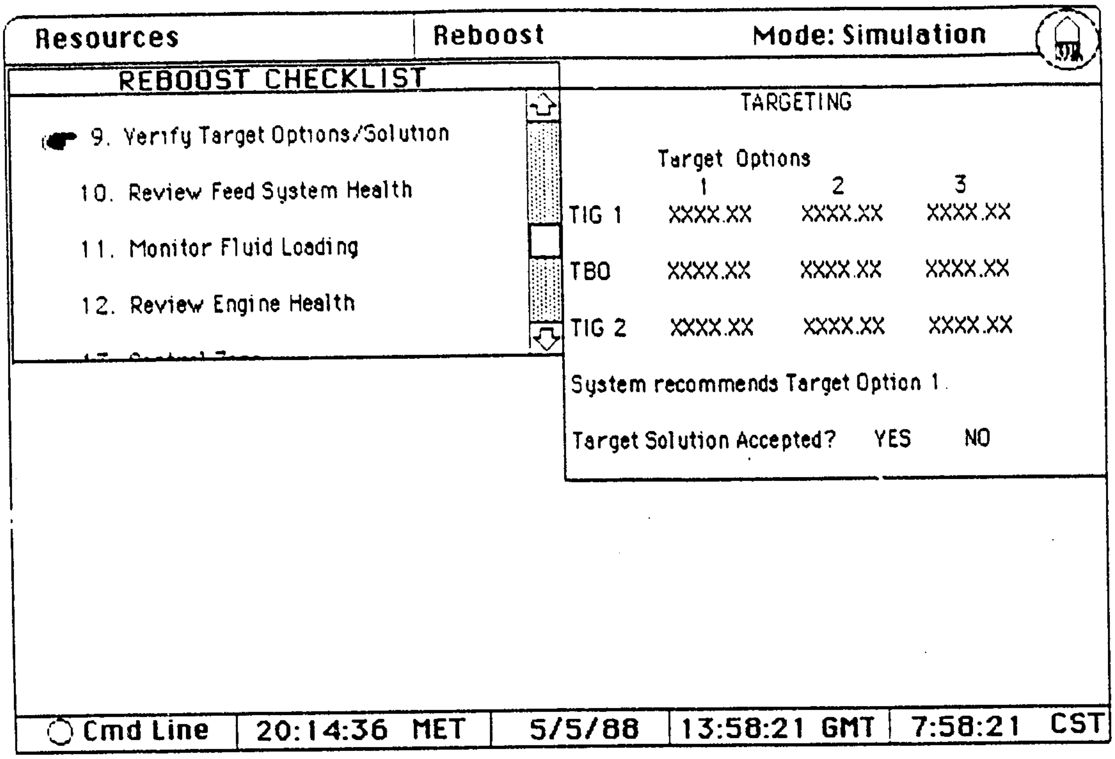
11. Monitor Fluid Loading

HEATERS AND TEMPERATURES

Heaters and Temperatures are ok? YES NO

○ Cmd Line | 20:14:36 MET | 5/5/88 | 13:58:21 GMT | 7:58:21 CST

FIGURE 6–14 REBOOST STEP 8 — GAUGES FIL'.ED



REBOOST CHECKLIST

9. Verify Target Options/Solution

10. Review Feed System Health

11. Monitor Fluid Loading

12. Review Engine Health

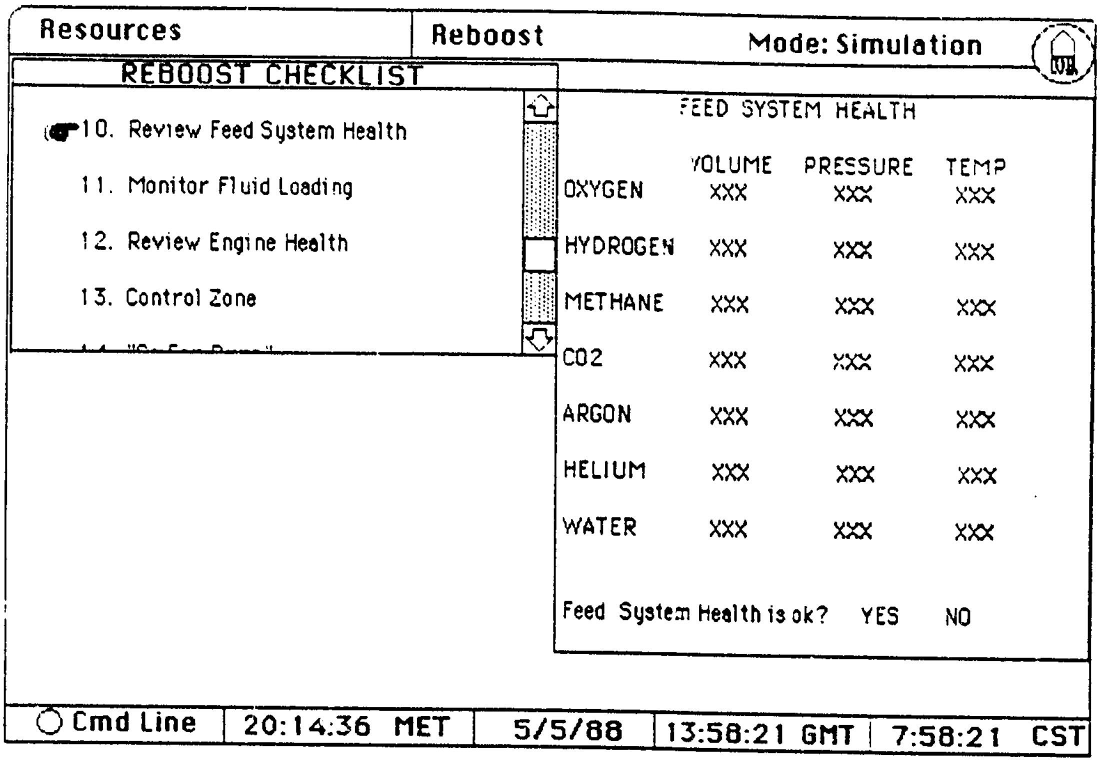
TARGETING

	Target Options 1	2	3
TIG 1	XXXX.XX	XXXX.XX	XXXX.XX
TBO	XXXX.XX	XXXX.XX	XXXX.XX
TIG 2	XXXX.XX	XXXX.XX	XXXX.XX

System recommends Target Option 1.

Target Solution Accepted? YES NO

FIGURE 6–15 REBOOST STEP 9



REBOOST CHECKLIST

10. Review Feed System Health

11. Monitor Fluid Loading

12. Review Engine Health

13. Control Zone

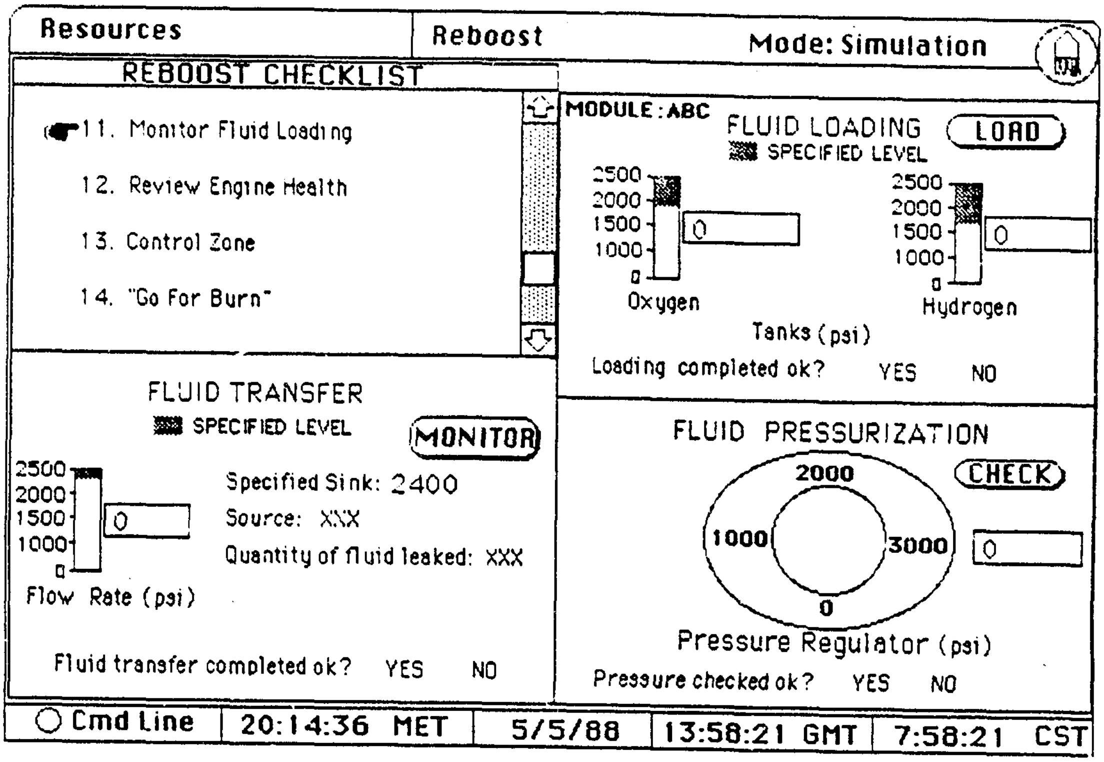
FEED SYSTEM HEALTH

	VOLUME	PRESSURE	TEMP
OXYGEN	XXX	XXX	XXX
HYDROGEN	XXX	XXX	XXX
METHANE	XXX	XXX	XXX
CO2	XXX	XXX	XXX
ARGON	XXX	XXX	XXX
HELIUM	XXX	XXX	XXX
WATER	XXX	XXX	XXX

Feed System Health is ok? YES NO

20:14:36 MET | 5/5/88 | 13:58:21 GMT | 7:58:21 CST

FIGURE 6–16 REBOOST STEP 10



11. Monitor Fluid Loading

12. Review Engine Health

13. Control Zone

14. "Go For Burn"

Oxygen: 0

Hydrogen: 0

Loading completed ok? YES NO

Specified Sink: 2400

Source: XXX

Quantity of fluid leaked: XXX

Flow Rate (psi): 0

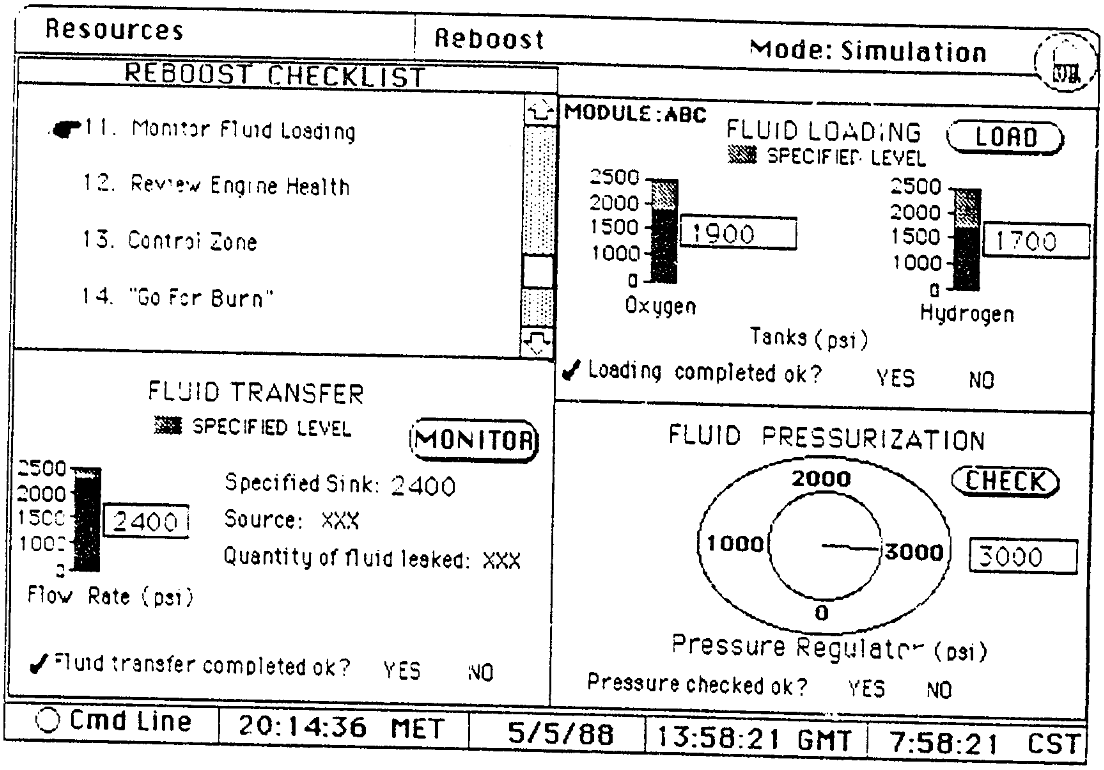
Fluid transfer completed ok? YES NO

Pressure Regulator (psi): 0

Pressure checked ok? YES NO

20:14:36 MET | 5/5/88 | 13:58:21 GMT | 7:58:21 CST

FIGURE 6-17 REBOOST STEP 11



REBOOST CHECKLIST

11. Monitor Fluid Loading

12. Review Engine Health

13. Control Zone

14. "Go For Burn"

MODULE: ABC

FLUID LOADING

▨ SPECIFIED LEVEL

Oxygen: 1900

Hydrogen: 1700

✓ Loading completed ok? YES NO

FLUID TRANSFER

▨ SPECIFIED LEVEL

2400

Specified Sink: 2400

Source: XXX

Quantity of fluid leaked: XXX

✓ Fluid transfer completed ok? YES NO

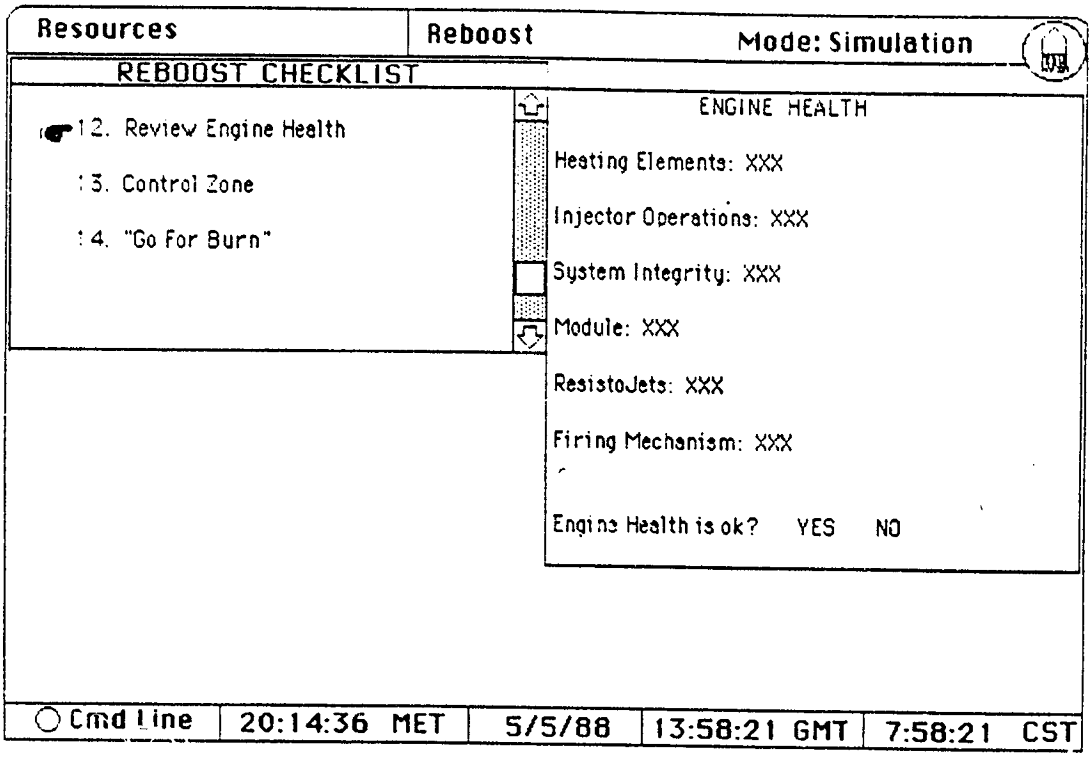
FLUID PRESSURIZATION

3000

Pressure checked ok? YES NO

20:14:36 MET	5/5/88	13:58:21 GMT	7:58:21 CST

FIGURE 6–18 REBOOST STEP 11 — GAUGES FILLED



REBOOST CHECKLIST

12. Review Engine Health

13. Control Zone

14. "Go For Burn"

ENGINE HEALTH

Heating Elements: XXX

Injector Operations: XXX

System Integrity: XXX

Module: XXX

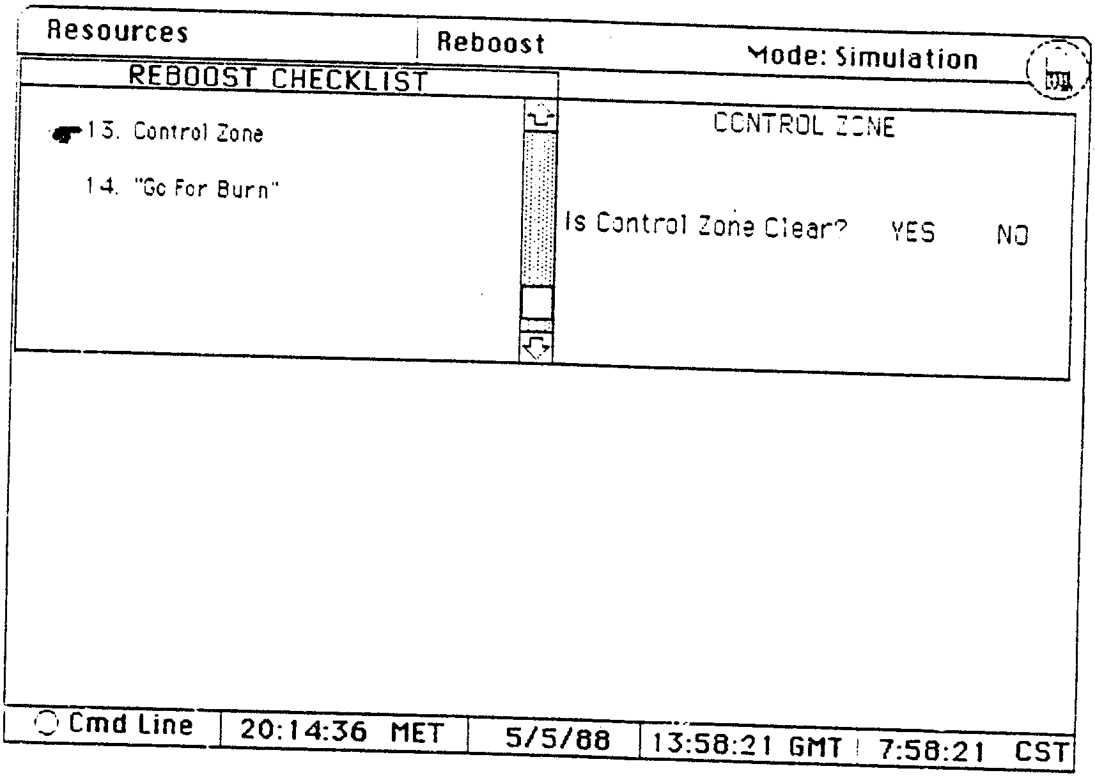
ResistoJets: XXX

Firing Mechanism: XXX

Engine Health is ok? YES NO

20:14:36 MET | 5/5/88 | 13:58:21 GMT | 7:58:21 CST

FIGURE 6–19 REBOOST STEP 12



REBOOST CHECKLIST

13. Control Zone

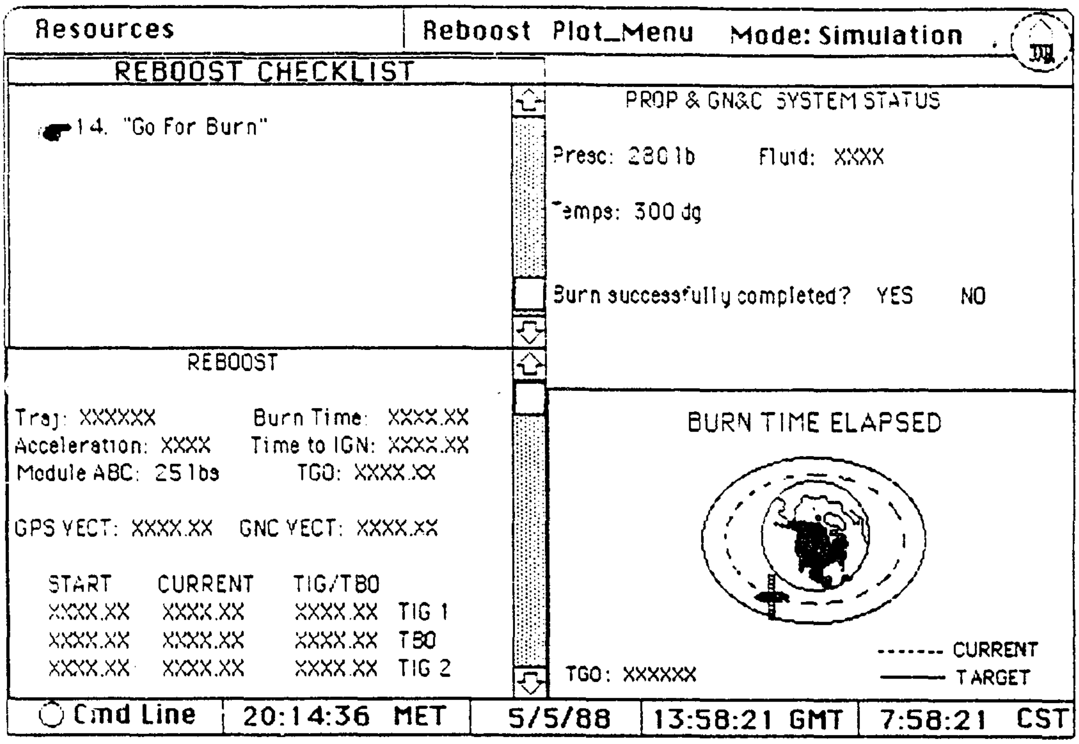
14. "Go For Burn"

CONTROL ZONE

Is Control Zone Clear? YES NO

20:14:36 MET | 5/5/88 | 13:58:21 GMT | 7:58:21 CST

FIGURE 6–20 REBOOST STEP 13



REBOOST CHECKLIST

14. "Go For Burn"

PROP & GN&C SYSTEM STATUS

Presc: 280 lb   Fluid: XXXX

Temps: 300 dg

Burn successfully completed? YES NO

REBOOST

Traj: XXXXXX   Burn Time: XXXX.XX
Acceleration: XXXX   Time to IGN: XXXX.XX
Module ABC: 25 lbs   TGO: XXXX.XX

GPS VECT: XXXX.XX   GNC VECT: XXXX.XX

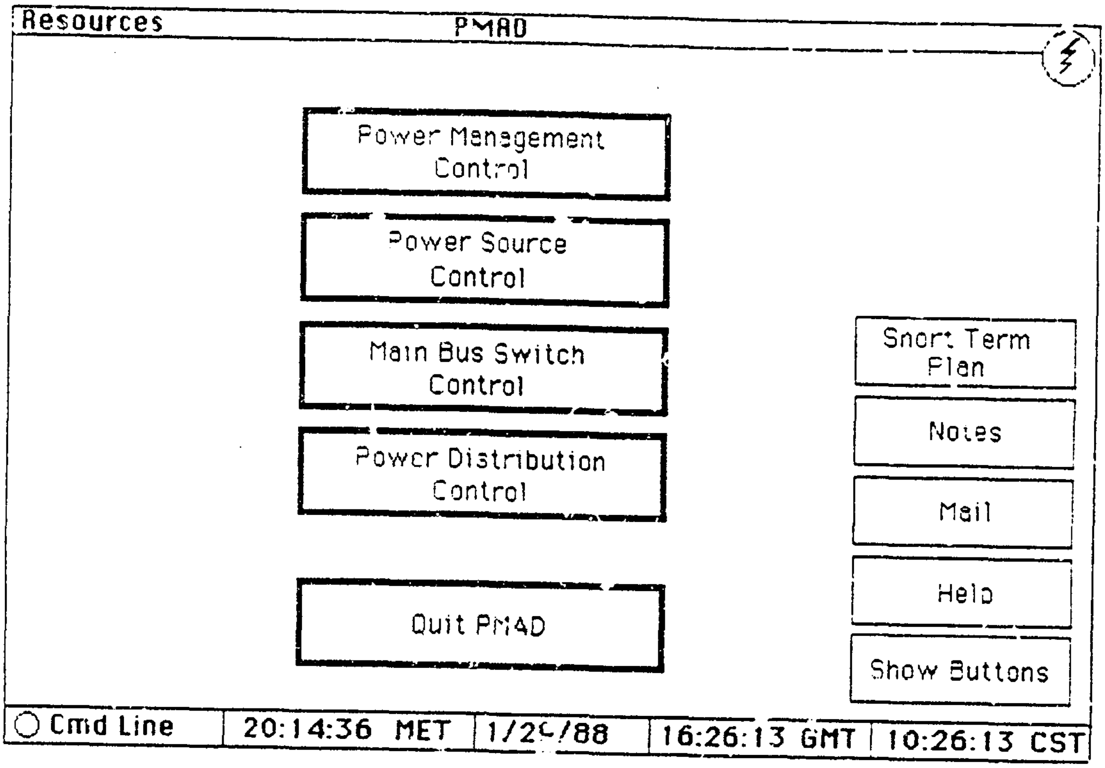
START	CURRENT	TIG/TBO	
XXXX.XX	XXXX.XX	XXXX.XX	TIG 1
XXXX.XX	XXXX.XX	XXXX.XX	TBO
XXXX.XX	XXXX.XX	XXXX.XX	TIG 2

BURN TIME ELAPSED

TGO: XXXXXX

20:14:36 MET | 5/5/88 | 13:58:21 GMT | 7:58:21 CST

FIGURE 6–21 REBOOST STEP 14



20:14:36 MET | 1/29/88 | 16:26:13 GMT | 10:26:13 CST

FIGURE 6-22 POWER MANAGEMENT AND DISTRIBUTION MAIN DISPLAY

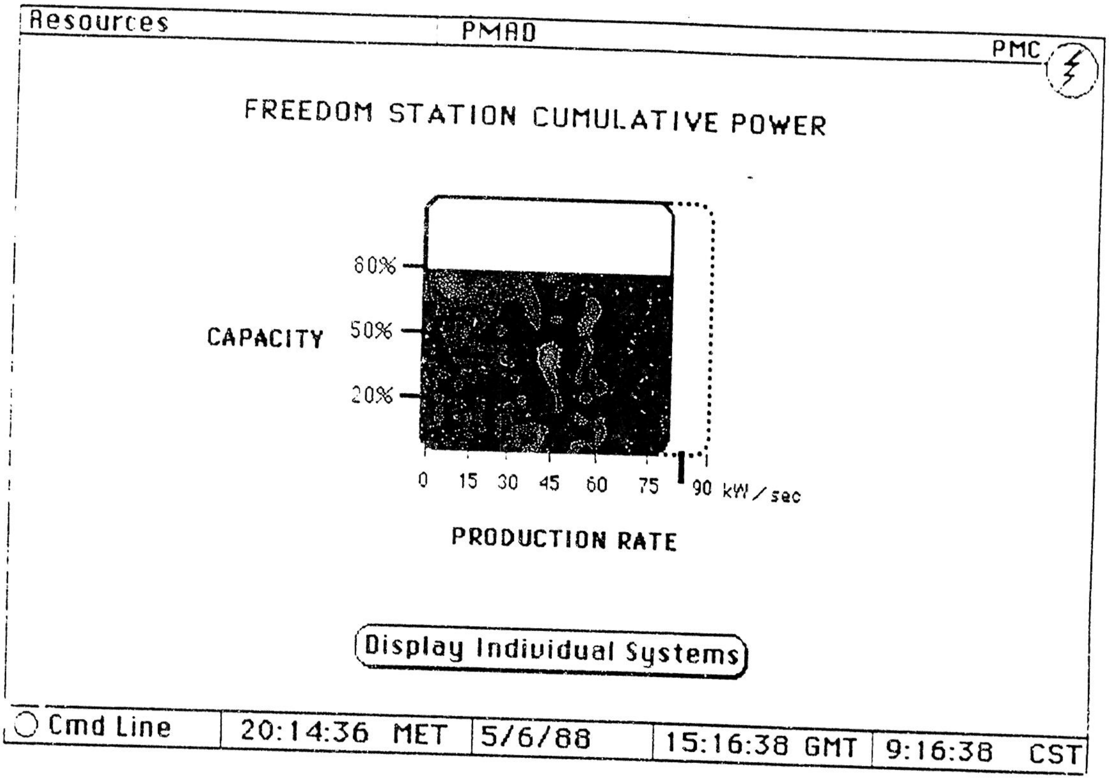


FIGURE 6–23 CUMULATIVE POWER AVAILABLE

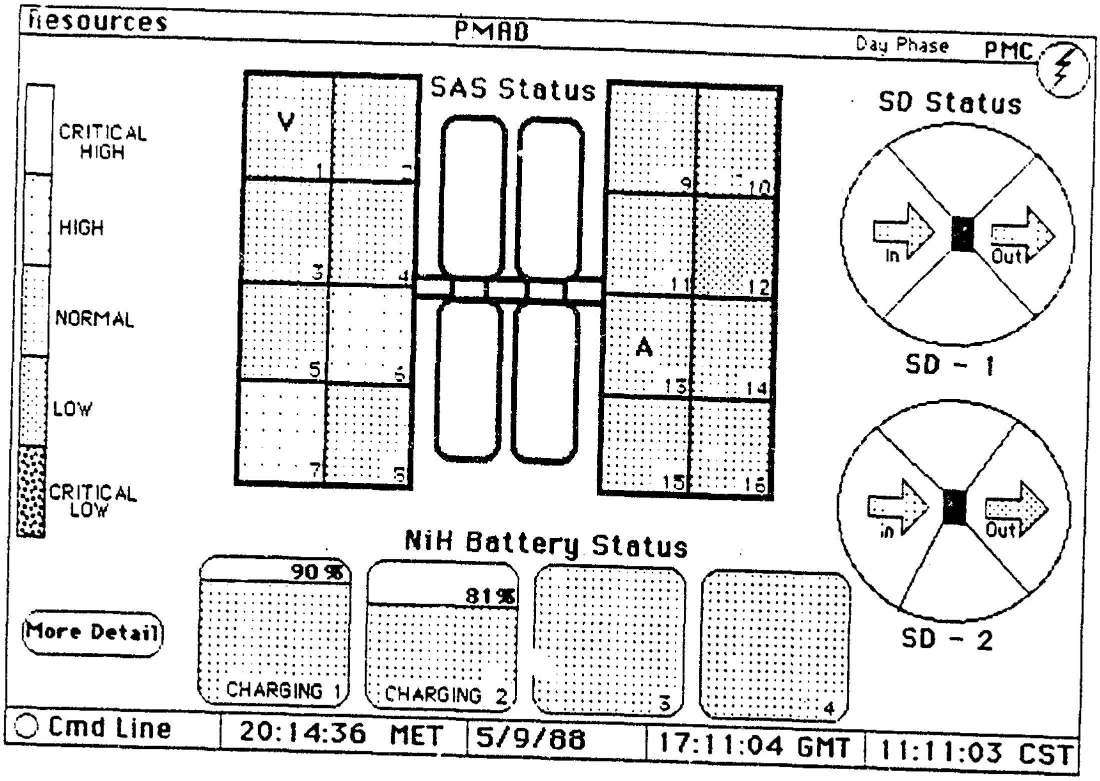
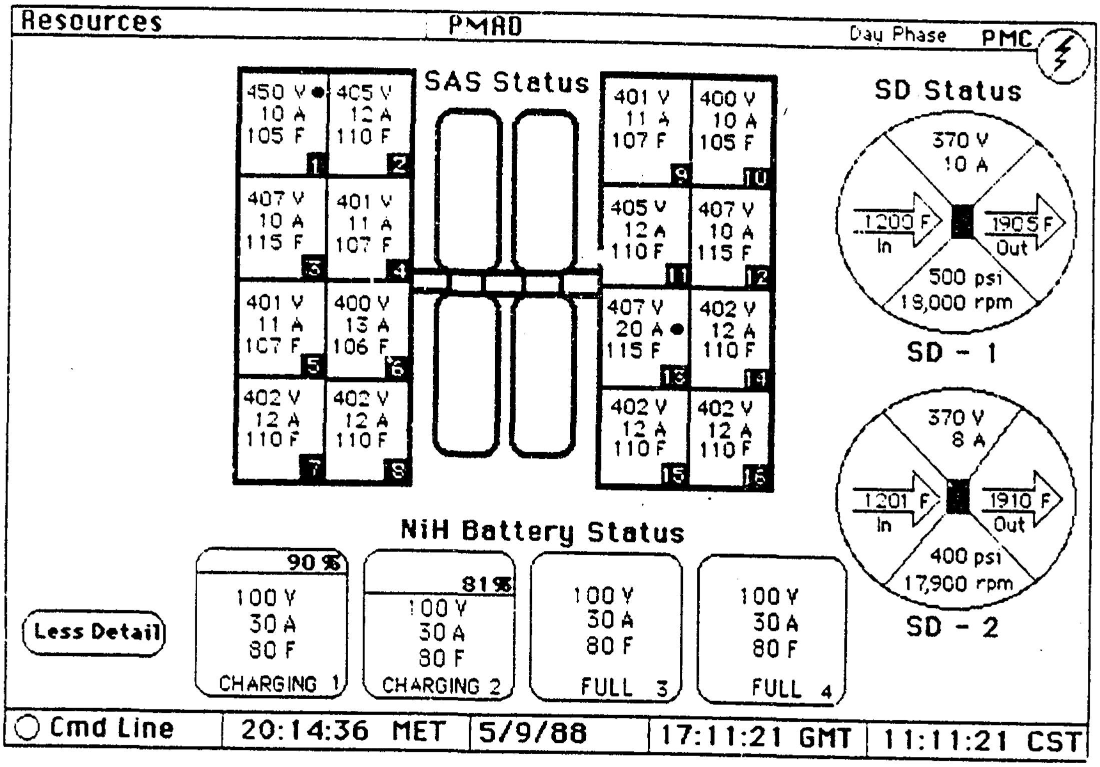


FIGURE 6–24 POWER SYSTEMS STATUS



Resources | PMAD | Day Phase | PMC

SAS Status

450 V ● 10 A 105 F 1	405 V 12 A 110 F 2
407 V 10 A 115 F 3	401 V 11 A 107 F 4
401 V 11 A 107 F 5	400 V 13 A 106 F 6
402 V 12 A 110 F 7	402 V 12 A 110 F 8

401 V 11 A 107 F 9	400 V 10 A 105 F 10
405 V 12 A 110 F 11	407 V 10 A 115 F 12
407 V 20 A ● 115 F 13	402 V 12 A 110 F 14
402 V 12 A 110 F 15	402 V 12 A 110 F 16

SD Status

SD - 1: 370 V, 10 A; In 1200 F; Out 1905 F; 500 psi, 18,000 rpm

SD - 2: 370 V, 8 A; In 1201 F; Out 1910 F; 400 psi, 17,900 rpm

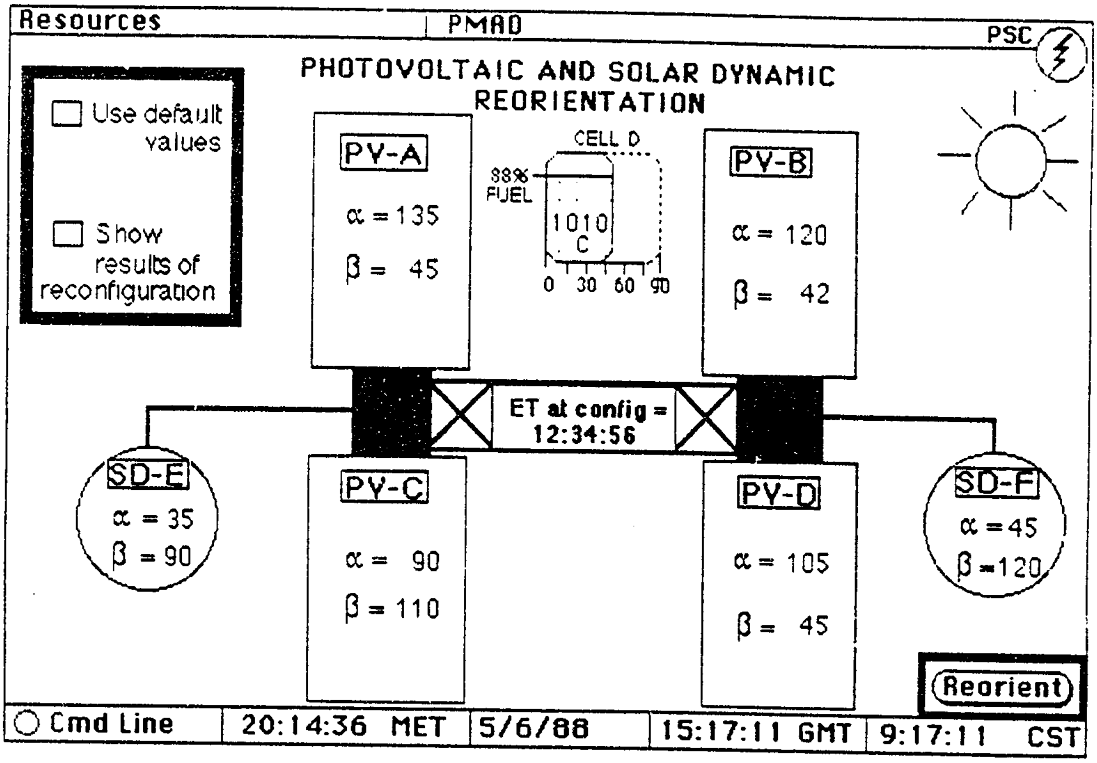
NiH Battery Status

90 %	81 %		
100 V	100 V	100 V	100 V
30 A	30 A	30 A	30 A
80 F	80 F	80 F	80 F
CHARGING 1	CHARGING 2	FULL 3	FULL 4

Less Detail

○ Cmd Line | 20:14:36 MET | 5/9/88 | 17:11:21 GMT | 11:11:21 CST

FIGURE 6–25 DETAILED SYSTEM STATUS



PHOTOVOLTAIC AND SOLAR DYNAMIC REORIENTATION

PV-A α = 135 β = 45

PV-B α = 120 β = 42

PV-C α = 90 β = 110

PV-D α = 105 β = 45

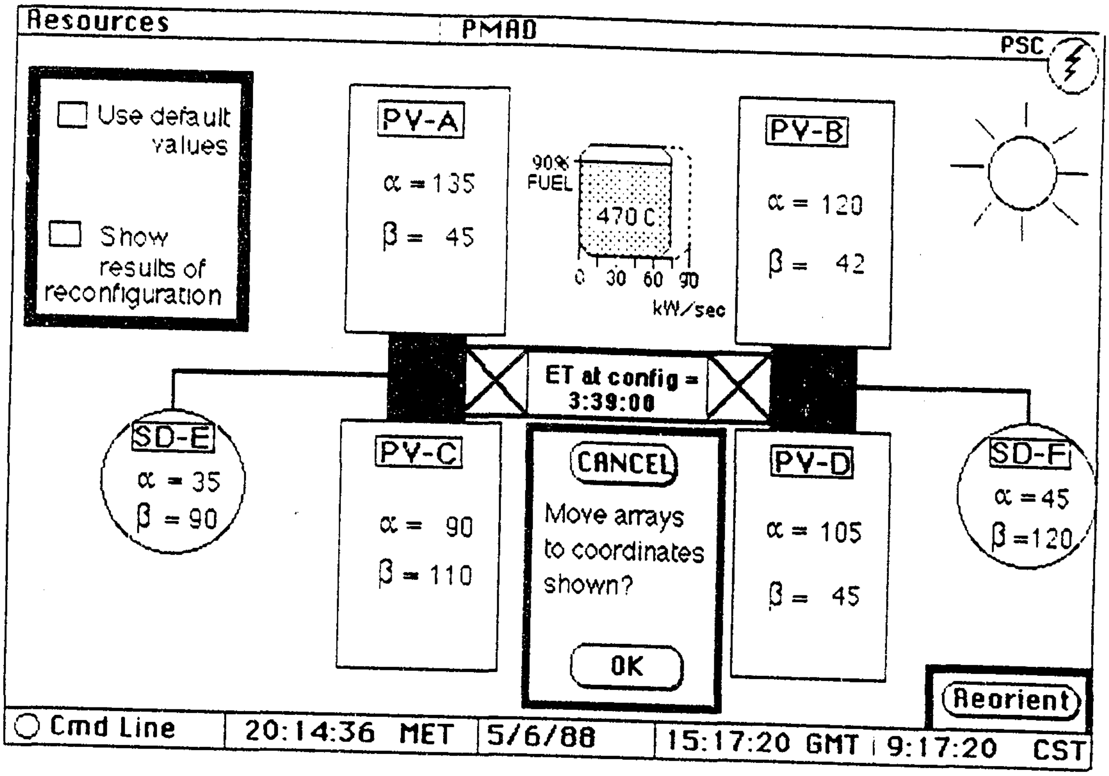
SD-E α = 35 β = 90

SD-F α = 45 β = 120

ET at config = 12:34:56

20:14:36 MET | 5/6/88 | 15:17:11 GMT | 9:17:11 CST

FIGURE 6–26 SOLAR ARRAY SECTION AND SOLAR DYNAMIC ORIENTATION



□ Use default values

□ Show results of reconfiguration

PV-A $\alpha = 135$ $\beta = 45$

PV-B $\alpha = 120$ $\beta = 42$

PV-C $\alpha = 90$ $\beta = 110$

PV-D $\alpha = 105$ $\beta = 45$

SD-E $\alpha = 35$ $\beta = 90$

SD-F $\alpha = 45$ $\beta = 120$

ET at config = 3:39:00

Move arrays to coordinates shown?

20:14:36 MET | 5/6/88 | 15:17:20 GMT | 9:17:20 CST

FIGURE 6–27 VERIFICATION SCREEN FOR REORIENTATION

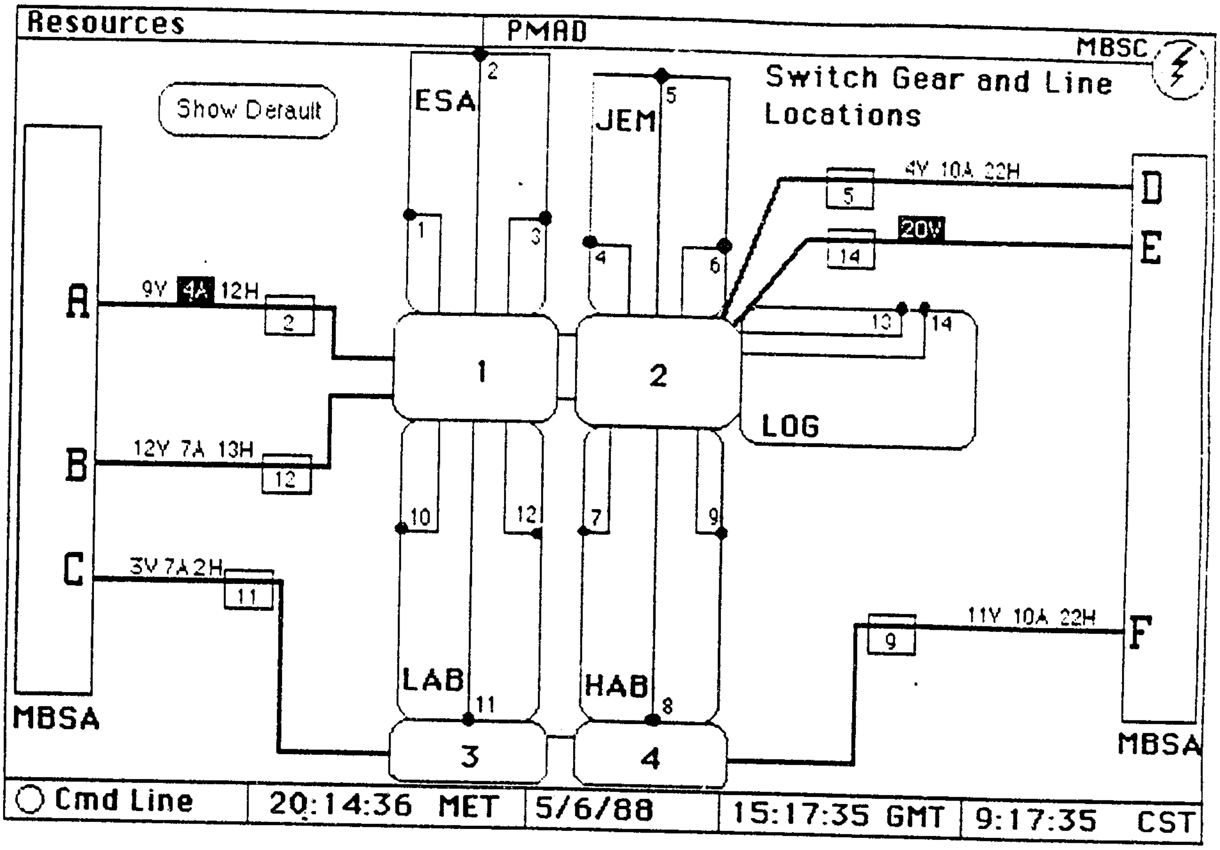
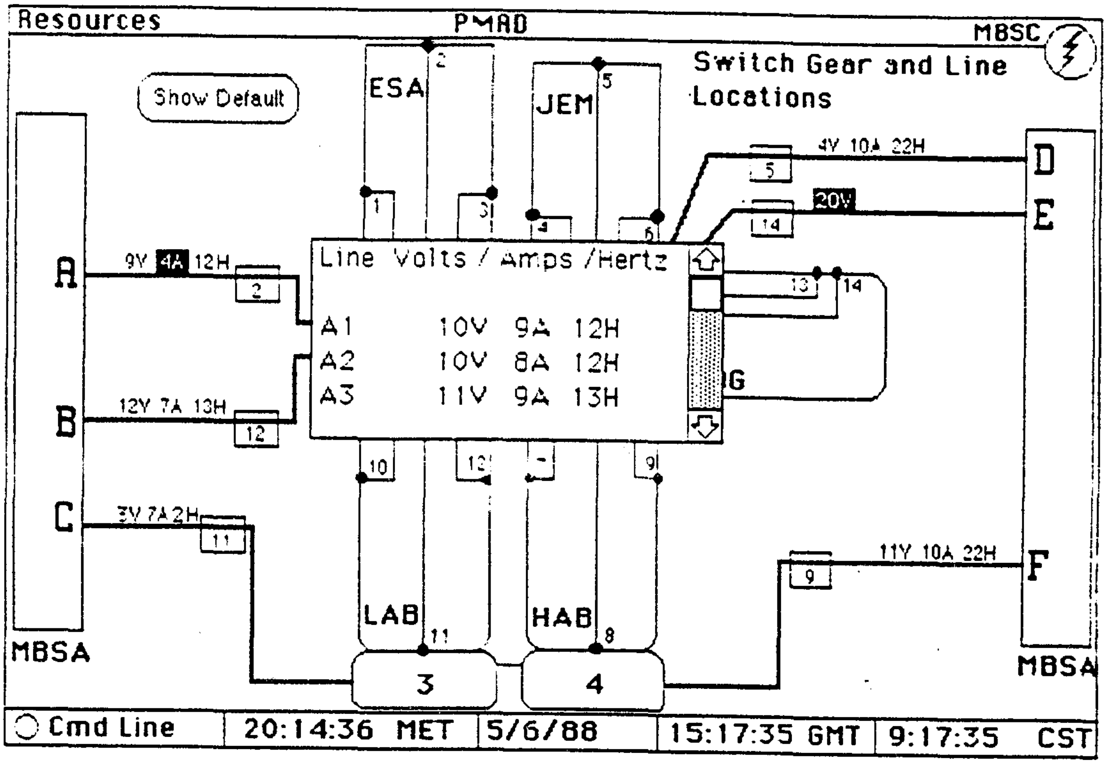


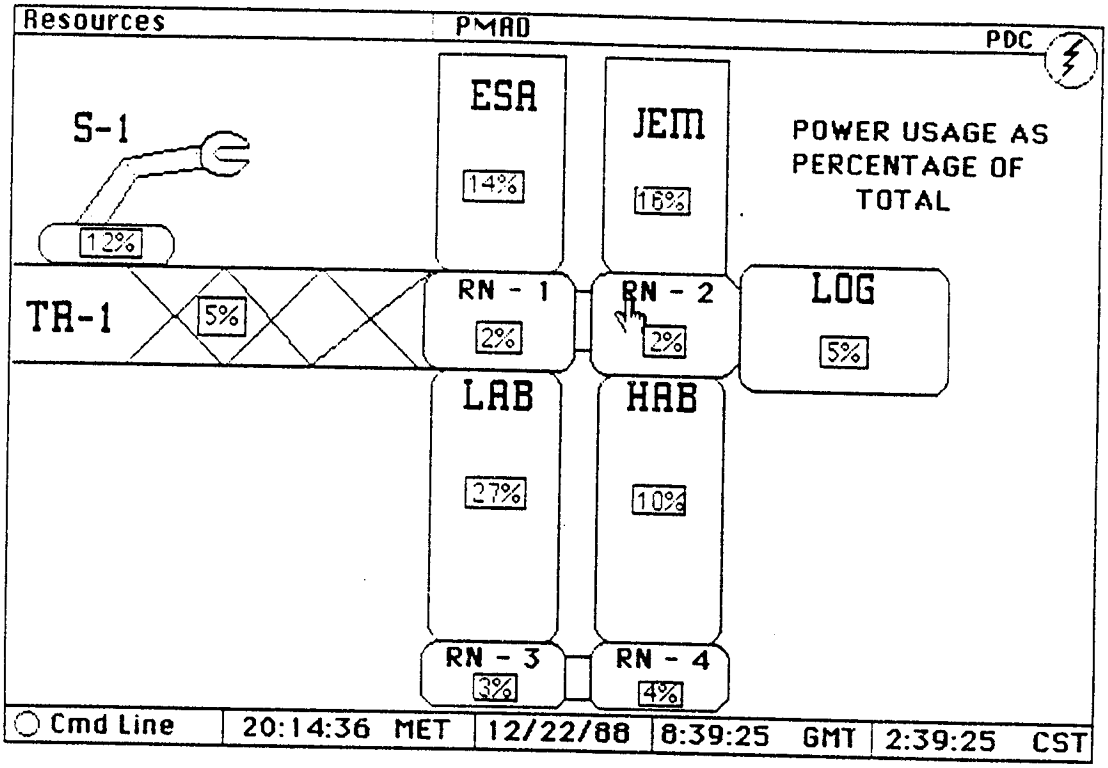
FIGURE 6–28 SWITCH GEAR AND POWER LINE LAYOUT



Line	Volts / Amps / Hertz		
A1	10V	9A	12H
A2	10V	8A	12H
A3	11V	9A	13H

○ Cmd Line | 20:14:36 MET | 5/6/88 | 15:17:35 GMT | 9:17:35 CST

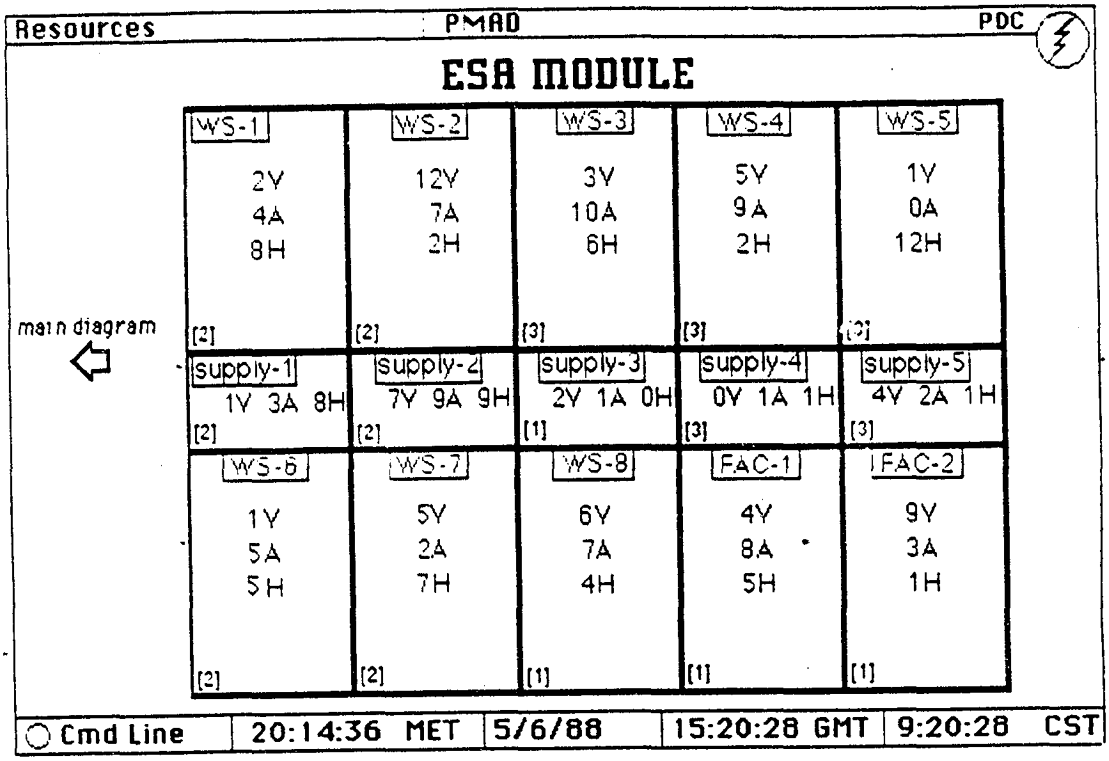
FIGURE 6–29 POP–UP DEFAULT VALUES FOR MAIN BUS SWITCH CONTROL



POWER USAGE AS PERCENTAGE OF TOTAL

20:14:36 MET | 12/22/88 | 8:39:25 GMT | 2:39:25 CST

FIGURE 6–30 POWER USAGE

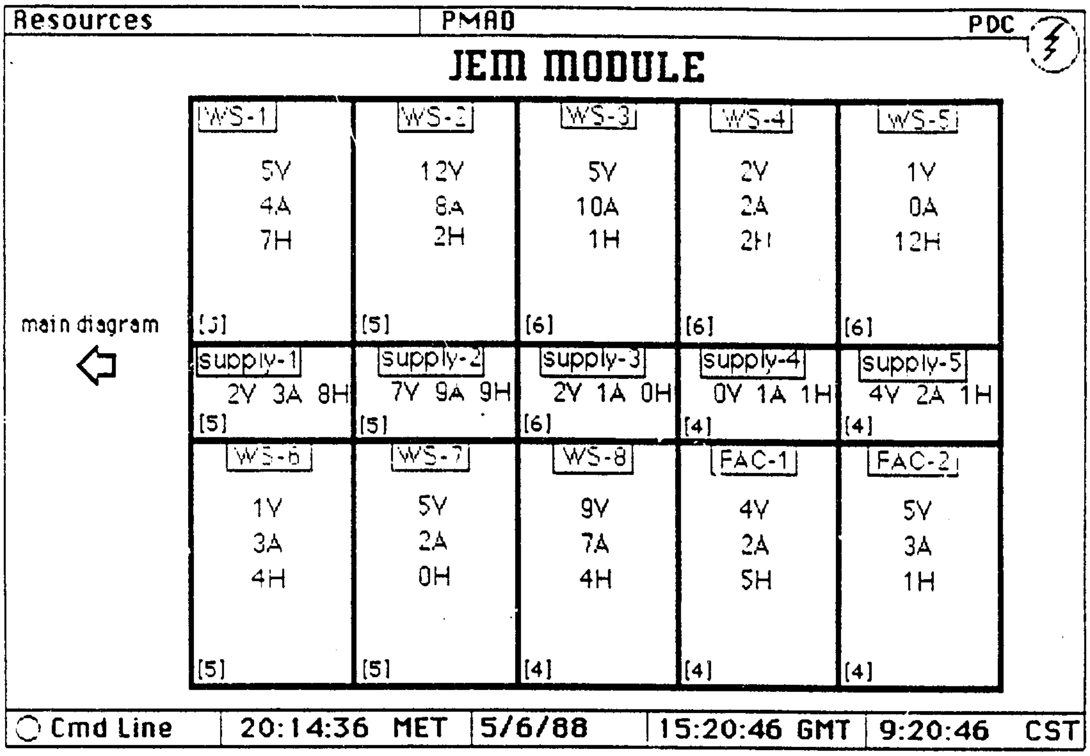


ESA MODULE

WS-1 2V 4A 8H [2]	WS-2 12V 7A 2H [2]	WS-3 3V 10A 6H [3]	WS-4 5V 9A 2H [3]	WS-5 1V 0A 12H [3]
supply-1 1V 3A 8H [2]	supply-2 7V 9A 9H [2]	supply-3 2V 1A 0H [1]	supply-4 0V 1A 1H [3]	supply-5 4V 2A 1H [3]
WS-6 1V 5A 5H [2]	WS-7 5V 2A 7H [2]	WS-8 6V 7A 4H [1]	FAC-1 4V 8A 5H [1]	FAC-2 9V 3A 1H [1]

○ Cmd Line | 20:14:36 MET | 5/6/88 | 15:20:28 GMT | 9:20:28 CST

FIGURE 6–31 DETAILED LAYOUT FOR EUROPEAN SPACE AGENCY MODULE

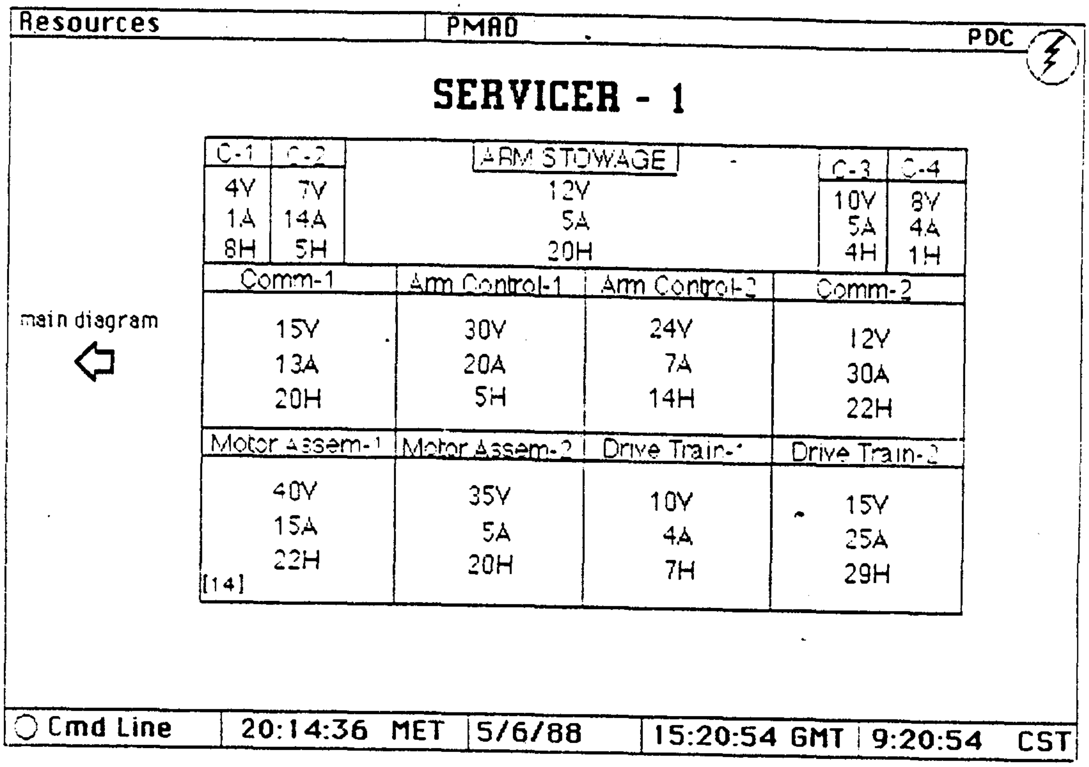


JEM MODULE

WS-1 5Y 4A 7H [J]	WS-2 12Y 8A 2H [5]	WS-3 5Y 10A 1H [6]	WS-4 2Y 2A 2H [6]	WS-5 1Y 0A 12H [6]
supply-1 2Y 3A 8H [5]	supply-2 7Y 9A 9H [5]	supply-3 2Y 1A 0H [6]	supply-4 0Y 1A 1H [4]	supply-5 4Y 2A 1H [4]
WS-6 1Y 3A 4H [5]	WS-7 5Y 2A 0H [5]	WS-8 9Y 7A 4H [4]	FAC-1 4Y 2A 5H [4]	FAC-2 5Y 3A 1H [4]

○ Cmd Line | 20:14:36 MET | 5/6/88 | 15:20:46 GMT | 9:20:46 CST

FIGURE 6–32 DETAILED JAPANESE EXPERIMENT MODULE LAYCUT



SERVICER - 1

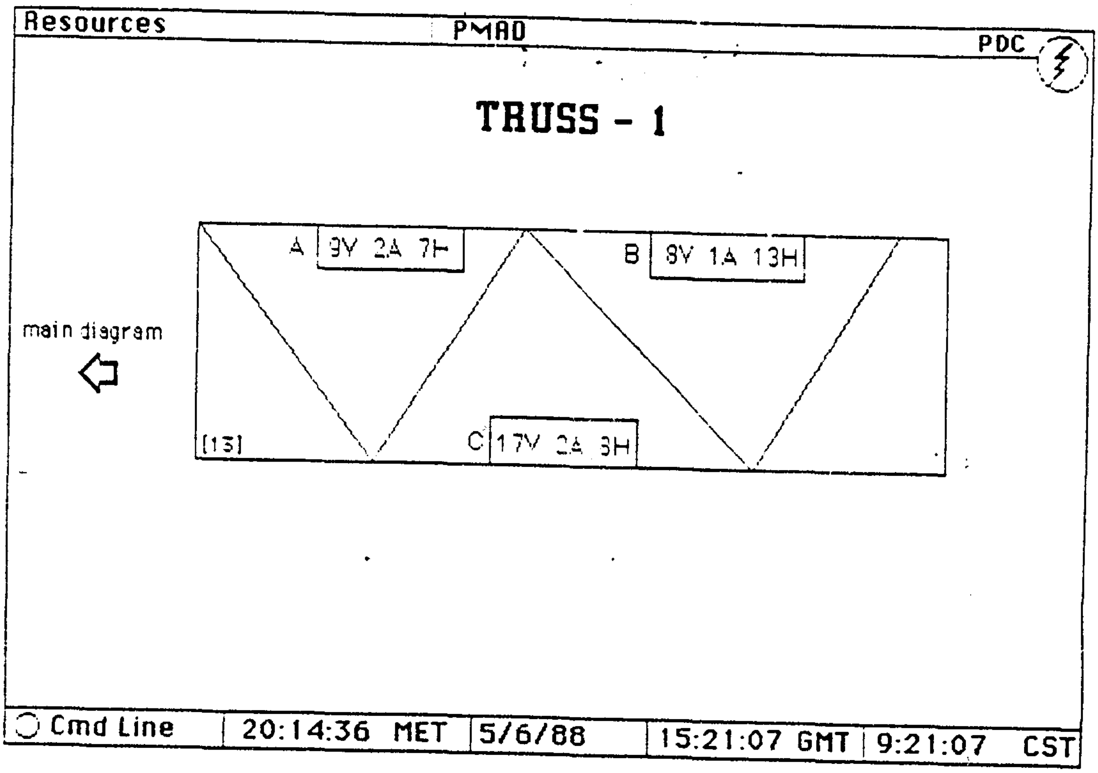
C-1	C-2	ARM STOWAGE	C-3	C-4
4V 1A 8H	7V 14A 5H	12V 5A 20H	10V 5A 4H	8V 4A 1H

Comm-1	Arm Control-1	Arm Control-2	Comm-2
15V 13A 20H	30V 20A 5H	24V 7A 14H	12V 30A 22H

Motor Assem-1	Motor Assem-2	Drive Train-1	Drive Train-2
40V 15A 22H [14]	35V 5A 20H	10V 4A 7H	15V 25A 29H

20:14:36 MET 5/6/88 15:20:54 GMT 9:20:54 CST

FIGURE 6–33 DETAILED LAYOUT FOR ROBOTIC SERVICER



TRUSS - 1

○ Cmd Line | 20:14:36 MET | 5/6/88 | 15:21:07 GMT | 9:21:07 CST

FIGURE 6–34 DETAILED DIAGRAM OF TRUSS ASSEMBLY–1

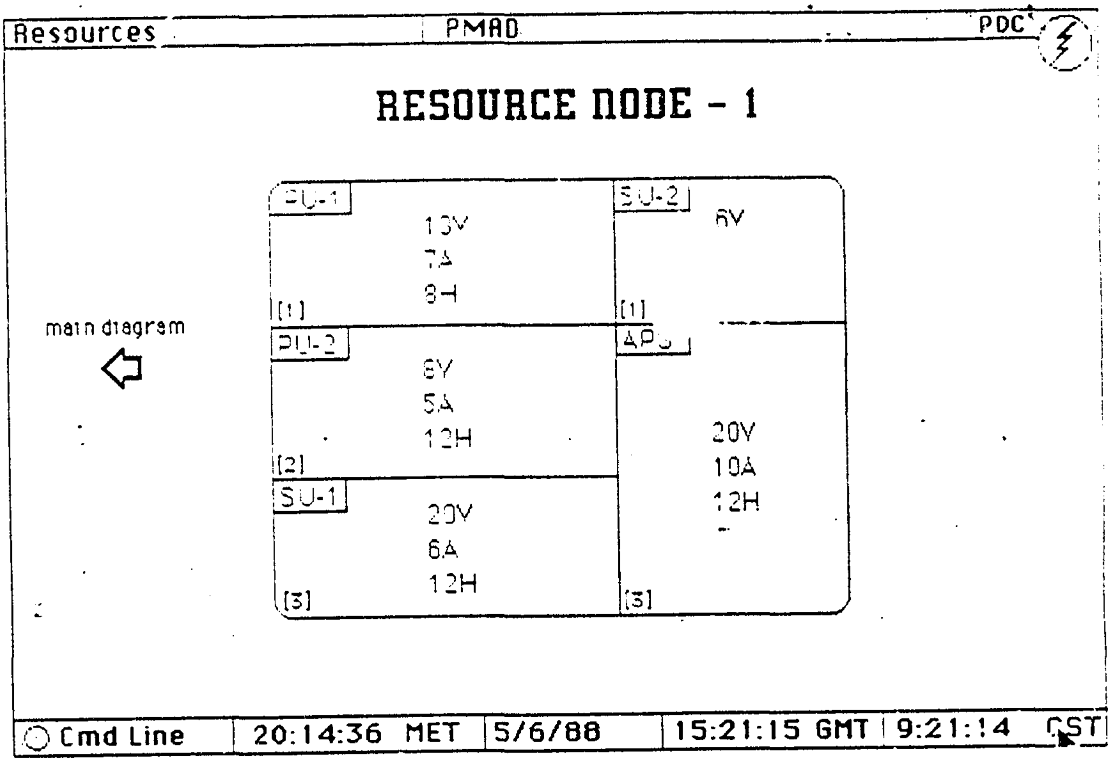
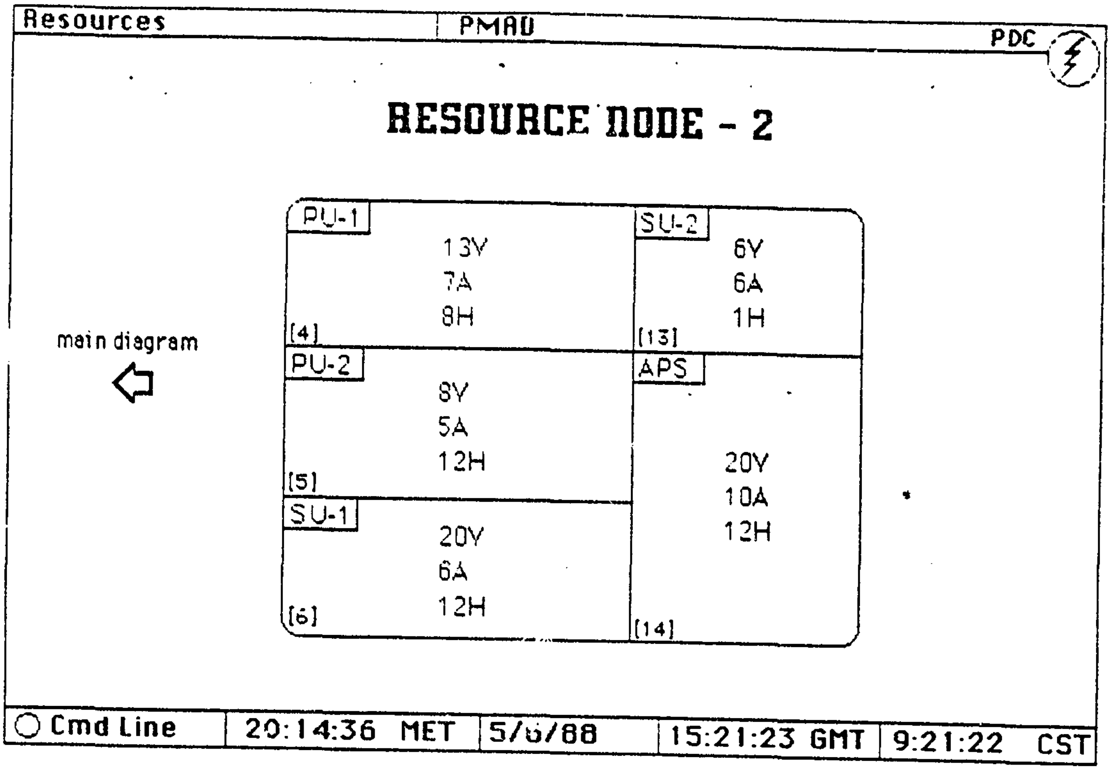


FIGURE 6–35 DETAILED DIAGRAM OF RESOURCE NODE–1



RESOURCE NODE - 2

PU-1
13V
7A
8H
[4]

SU-2
6V
6A
1H
[13]

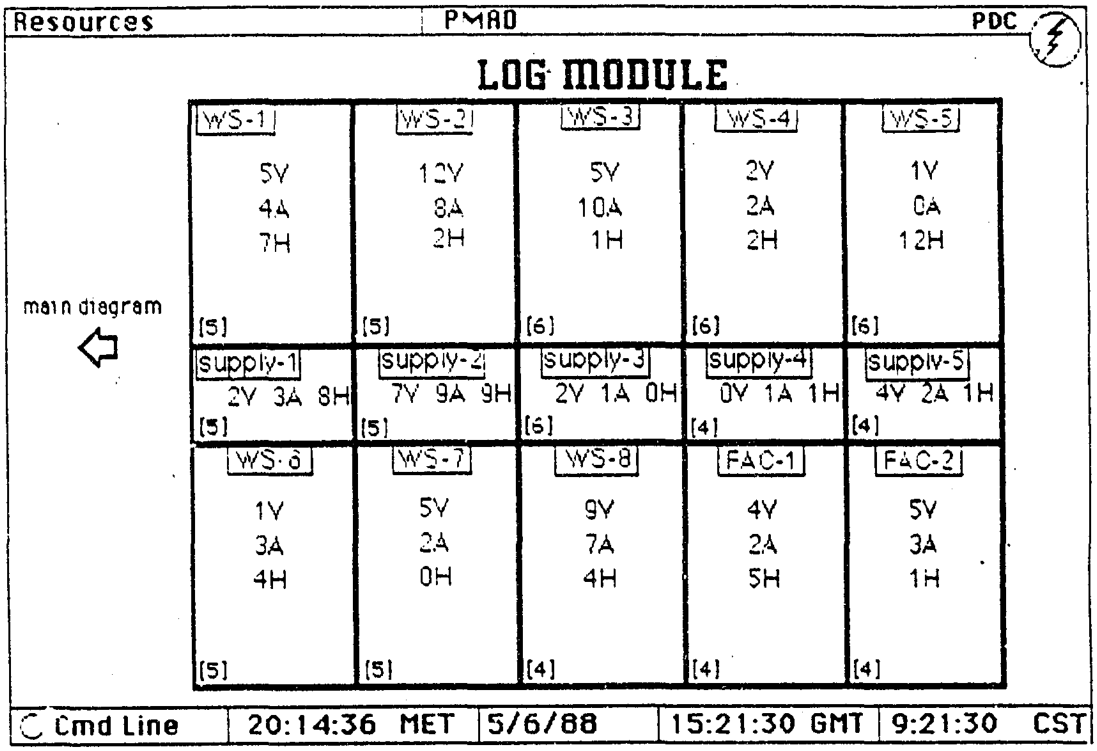
PU-2
8V
5A
12H
[5]

APS
20V
10A
12H
[14]

SU-1
20V
6A
12H
[6]

○ Cmd Line | 20:14:36 MET | 5/6/88 | 15:21:23 GMT | 9:21:22 CST

FIGURE 6–36 DETAILED DIAGRAM OF RESOURCE NODE–2



Resources | PMAD | PDC

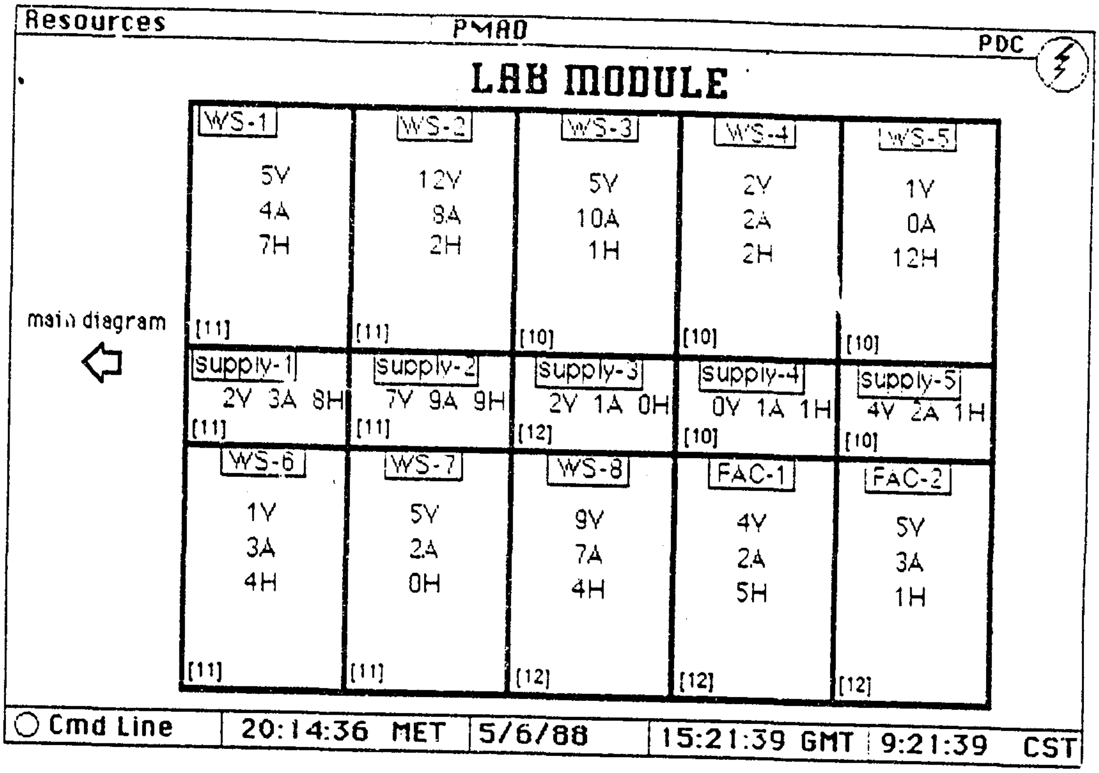
LOG MODULE

main diagram

WS-1 5V 4A 7H [5]	WS-2 12V 8A 2H [5]	WS-3 5V 10A 1H [6]	WS-4 2V 2A 2H [6]	WS-5 1V 0A 12H [6]
supply-1 2V 3A 8H [5]	supply-2 7V 9A 9H [5]	supply-3 2V 1A 0H [6]	supply-4 0V 1A 1H [4]	supply-5 4V 2A 1H [4]
WS-6 1V 3A 4H [5]	WS-7 5V 2A 0H [5]	WS-8 9V 7A 4H [4]	FAC-1 4V 2A 5H [4]	FAC-2 5V 3A 1H [4]

Cmd Line | 20:14:36 MET | 5/6/88 | 15:21:30 GMT | 9:21:30 CST

FIGURE 6–37 DETAILED LAYOUT OF LOG MODULE



Resources PMAD PDC

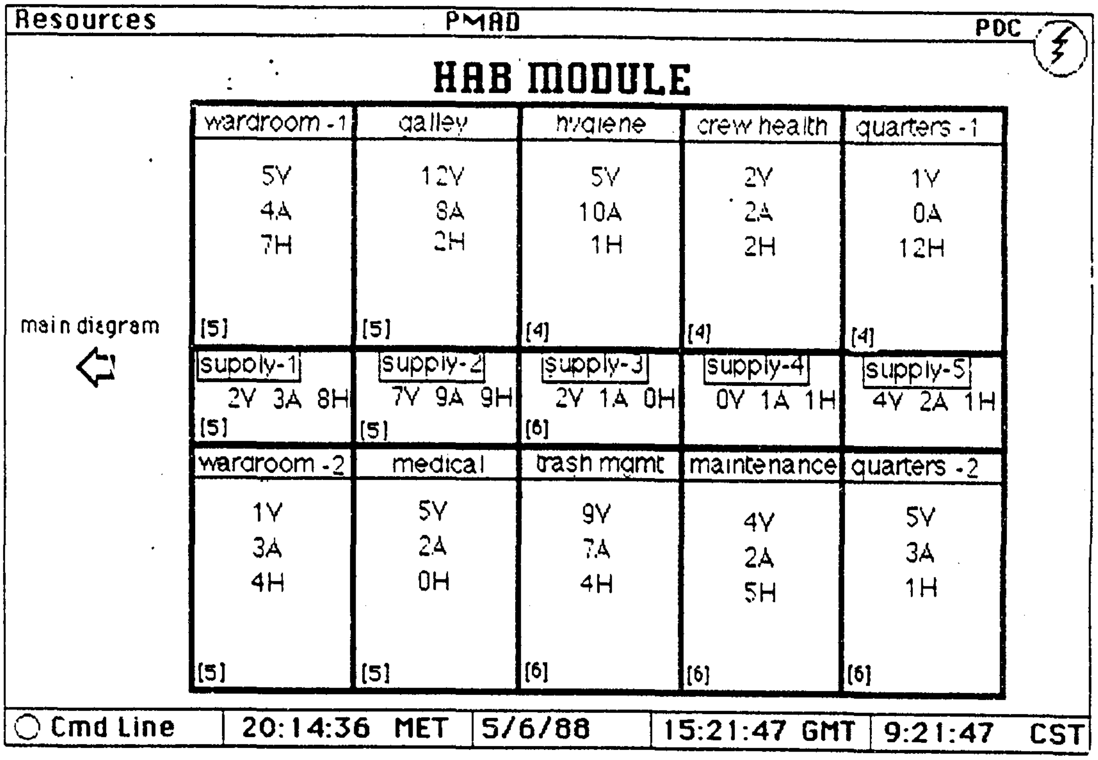
LAB MODULE

main diagram ⇦

WS-1 5V 4A 7H [11]	WS-2 12V 8A 2H [11]	WS-3 5V 10A 1H [10]	WS-4 2V 2A 2H [10]	WS-5 1V 0A 12H [10]
supply-1 2V 3A 8H [11]	supply-2 7V 9A 9H [11]	supply-3 2V 1A 0H [12]	supply-4 0V 1A 1H [10]	supply-5 4V 2A 1H [10]
WS-6 1V 3A 4H [11]	WS-7 5V 2A 0H [11]	WS-8 9V 7A 4H [12]	FAC-1 4V 2A 5H [12]	FAC-2 5V 3A 1H [12]

○ Cmd Line | 20:14:36 MET | 5/6/88 | 15:21:39 GMT | 9:21:39 CST

FIGURE 6–38 DETAILED LAYOUT OF LAB MODULE



Resources | PMAD | PDC

HAB MODULE

main diagram

wardroom -1	galley	hygiene	crew health	quarters -1
5V 4A 7H [5]	12V 8A 2H [5]	5V 10A 1H [4]	2V 2A 2H [4]	1V 0A 12H [4]
supply-1 2V 3A 8H [5]	supply-2 7V 9A 9H [5]	supply-3 2V 1A 0H [6]	supply-4 0V 1A 1H	supply-5 4V 2A 1H
wardroom -2	medical	trash mgmt	maintenance	quarters -2
1V 3A 4H [5]	5V 2A 0H [5]	9V 7A 4H [6]	4V 2A 5H [6]	5V 3A 1H [6]

○ Cmd Line | 20:14:36 MET | 5/6/88 | 15:21:47 GMT | 9:21:47 CST

FIGURF 6–39 DETAILED LAYOUT FOR LAB MODULE

APPENDIX A ABBREVIATIONS AND ACRONYMS

BC	Ballistocardiography
C&T	Communication and Tracking
C&W	Caution and Warning
CAD	Computer Aided Design
CAE	Computer Aided Engineering
CMG	Control Moment Gyro
cr	Correct Rejections
CRT	Cathode Ray Tube
CSA	Canadian Space Agency
DMS	Data Management System
ECLS	Environmental Control and Life Support
EMS	Electronic Mail System
EP	Execute Plan
ESA	European Space Agency
EVA	Extravehicular Activity
fa	False Alarm
FAA	Federal Aviation Administration
GN&C	Guidance Navigation and Control
HCI	Human–Computer Interface
HCIG	Human–Computer Interface Guide
HOL	High Order Language
IP	Increment Plan
IFM	In Flight Maintenance
IWG	Interface Working Group
JSC	Johnson Space Center

LESC	Lockheed Engineering and Sciences Company
MBSC	Main Bus Switch Control
MET	Mission–Elapsed Time
MSC	Mobile Servicing Center
MSCS	Mobile Servicing Center System
MPS	Mission Planning System
MSE	Mean Square Error
MSIS	Man–Systems Integration Standards
NASA	National Aeronautics and Space Administration
NASCOM	NASA Realtime Communications Network
NASDA	National Space Development Agency of Japan
NSTS	National Space Transportation System
OMGA	Operations Management Ground Application
OMS	Operations Management System
OPS	Operations Planning System
PCC	Platform Control Center
PDC	Power Distribution Control
PDRD	Program Definition and Requirements Document
PI	Principal Investigator
PMAD	Power Management and Distribution
PMC	Power Management Control
POIC	Payload Operations Integration Center
PROP	Propulsion
PSC	Power Source Control
PSCN	Program Support Communications Network
PV	Photovoltaic

RCS	Reaction Control System (Subsystem)
SSCC	Space Station Control Center
SSE	Software Support Environment
SSF	Space Station Freedom
SSFP	Space Station Freedom Program
SSMB	Space Station Manned Base
STP	Short Term Plan
STS	Space Transportation System
SD	Solar Dynamic
TBD	To Be Determined
TCS	Thermal Control System
TMIS	Technical and Management Information System
TOP	Tactical Operations Plan
UIL	User Interface Language
USE	User Support Environment

APPENDIX B GLOSSARY

NOTE: Selected words from the glossary are capitalized in paragraph 4.0.

ACTIVE WINDOW (See WINDOW.)

Commands issued by the user are directed to an active window. Typically, this means that an active window (a) is currently receiving input from the user, (b) has last received input from the user, or (c) has been readied for input through the user's explicit action. In any event, the user is generally said to be "working in" the active window (processing a document, controlling a system, entering data, etc.).

ALPHANUMERIC CODE

A set of letters and/or numbers used to identify a group of data (e.g., in a table or on a statistical graph).

ATTRIBUTE

A characteristic of an element; especially in the context of HCIG as it applies to data display coding or representation (e.g., color, font).

AUDITORY PRESENTATION

A means of presenting information including verbal information presented by either speech (recorded or electronically created) or nonspeech sounds (e.g., bells, whistles, beeps).

BUTTON

A defined control region on the display screen which, when selected, causes some action.

CHARACTER WIDTH

The horizontal distance between a character's origin (a point on the base line used as a reference location) and the next character's origin.

CLEAR

A system function which removes the current selection but doesn't put it into the temporary editing buffer. A copy is retained, accessible immediately by the Undo command.

CLICK

An input device button–down event, distinct from cursor positioning, for the actual entry (enabling, activation) of a designated position.

CLIPBOARD

See TEMPORARY EDITING BUFFER.

CLOSED WINDOW (See WINDOW.)

Requires some action by the user in order to gain perceptual and functional access to the window. For example, a user may select and open an icon that represents a window or, in contrast, the user might input a UIL command to open a specific window.

CODING

Is used for highlighting (i.e., to attract a user's attention to part of a display); as a perceptual indicator of a data group; or to symbolize a state or attribute of an object (e.g., to show a temperature level or for warning purposes).

COMMAND

User–initiated messages to the system used to specify desired functions.

COMMAND AREA

An area of the screen presented at the user's request into which User Interface Language entries may be typed and then entered to the system as commands.

CONDITIONAL CUES

Provide the user with information about the rules that operate under the current conditions.

CONTROLLED STUDY

An investigation in which an independent variable (e.g., the type of display format or different dialogue techniques) is directly and systematically manipulated and the effects of manipulating that variable are measured.

COPY

A system function that puts a copy of the selection into the temporary editing buffer without disrupting the original data.

CURSOR

A display structure that is used to indicate the position of the user's operation on the display. Cursors serve two different functions: placeholding and pointing. Placeholding involves showing the location of the immediately previous operation or the point at which the user has moved the cursor. Pointing involves indicating the user's position in relation to certain other display structures such as icons, menu bars and items, and scroll bars. The pointing cursor can be used to position the placeholder cursor.

CUT

A system function that removes the current selection from the screen and puts it into the temporary editing buffer, replacing the buffer's previous contents. Cut may be used to either delete or to move a selection.

DATA DISPLAY CODE

Code consisting of graphical objects that represent data in a graph, diagram, or map. An example of a data display code is the use of different shapes of objects to plot data from different groups.

DATA FORMS

A user interaction tool which can support data entry and human–computer dialogue. (See also FORMS.)

DATABASE

A collection of data that is stored for computer access.

DIRECT MANIPULATION CONTROL

Defined by the close temporal and physical relations between the movement of the control device and the cursor, or other screen–based follower (e.g., an icon or a window). Direct manipulation control devices include the mouse, the trackball, and pointing devices. In general, a direct manipulation device permits the user to move the cursor and to use the cursor to select a display structure (e.g., by clicking on a button on the device).

DIRECT MANIPULATION DIALOGUE

The user manipulates symbols in the display by directly interacting with the symbol. The direct manipulation is generally performed through the use of a display structure, such as a pointer, and a cursor control device, such as a mouse.

DISPLACEMENT JOYSTICK

(Also known as the isotonic joystick) Provides output that is proportional to and in the same direction as the displacement of the joystick from the center.

DISPLAY

Refers to a specific integrated, organized set of information in the form of computer output that is required to perform a task or a step in a task.

DISPLAY STRUCTURES

Information–presenting elements that are consistent in appearance and use across applications. Their functions include providing reference to the user's location, reminding the user what options are available, and providing a visible boundary for user actions.

DOUBLE–CLICK

Two input device clicks within a default of 700 ms of each other. This value should be user modifiable.

DRAG

The act of moving a selected screen element or cursor through parts of a display.

DYNAMIC DISPLAY

Contains screen structures which change one or more feature(s), e.g., numerical value, color, shape, or spatial location, in real time or near real time.

ENTER

An explicit user function that effects computer processing of user entries.

EXCERPT FILE

Similar to the temporary editing buffer but in addition to allowing the user to move data from one location to another it permits the user to perform a variety of functions on the data that a buffer does not.

FEEDBACK

Any system response to a user action. Implies acknowledgment of the user action.

FILE

A collection of data treated as a single unit for storage purposes.

FIXED FUNCTION KEYS

Keys which have a function that cannot be changed by the user or system and that remain constant between applications.

FOLLOWER

An object moved by direct manipulation; including pointing cursors, icons, windows or any selected object.

FORCE JOYSTICK

(Also called an isometric joystick or a pressure joystick) Provides output proportional to and in the same direction as the user's perceived applied force on a lever that does not move.

FORMS

A dialogue technique which presents category labels and requires the user to fill in the blanks.

FUNCTION

A software supported capability provided to a user to aid in task performance.

FUNCTION AREAS

Specific locations that are reserved for a specific purpose. Function areas can occur anywhere on the screen; that is, on the primary display or within a window which is part of the primary display.

FUNCTION KEY

A key whose activation results in the computer processing some programmed action.

GLOVE CONTROLLER

A light-weight glove–like device that transmits data records of arm, hand, finger shape, and position to a host computer.

GRAPHICAL DISPLAY

A display which provides a pictorial representation of an object or a set of data. Graphical displays include line, solid object, and perspective drawings; bar, pie, and line charts and graphs; scatterplots; displayed meters; flowcharts and schematic diagrams.

GRAPHICAL OBJECT

The graphically–displayed information of primary interest to the user (e.g., a data graph or a schematic diagram).

HANDCONTROLLER

As a general class, allows an operator to manipulate a small scale version of a larger and/or stronger system. These include teleoperators or remote handling equipment.

HARD COPY

A printed paper display output by the computer.

HELP

Information displayed at the user's request for on-line guidance. HELP provides both specific application or general system information.

HIERARCHICAL BRANCHING

A method of structuring menu items that are hierarchically related which provides for selection among alternatives without requiring the opening and closing of a series of menus; the entire hierarchy is contained in one menu.

HIGHLIGHTING

A means of directing the user's attention to a feature of the display. Highlighting methods include image reversal (reverse video), brightness/boldness contrast, color, underlining, blinking, flashing arrows, and changes in font.

HOT SPOT

An area of a display which acts as a hidden button, in other words, when a user selects the area defined as a "hot spot," a predefined action will occur. "Hot spots" are most often used to provide more information about an object.

ICON

Pictorial, pictographic, or other nonverbal representations of objects or actions.

INACTIVE WINDOW (See WINDOW.)

Windows perceptually and functionally available to the user (the user can see and obtain information from them) but not immediately available in the sense that the user must activate an inactive window before working in it.

INFORMATION AREA

An area containing general purpose information that would be helpful to all users, including the time, date, and version number of the application being used.

INSERT MODE

A data entry mode which allows the user to insert new text within existing text with no deletion of characters. If the cursor is placed within existing text, old characters are moved forward to allow insertion of the new characters.

INTERFACE

That aspect of the computer system apparent to the user in the form of displays, input/output devices, and the user's interaction with them.

KEYSTROKE COMMAND

A limited number of nonlinguistic keystrokes that define UIL commands. The keystrokes are often initiated by the simultaneous press of a key that signals a keystroke command and the first letter of a one word command. Another version of the keystroke command is the function key.

LABEL

Descriptor that is distinguishable from and helps to identify displayed screen structures.

LAYERING

A means of manipulating multiple windows which allows windows to overlap and obscure the contents of the covered windows.

LEGEND

An explanatory list of symbols or highlighting used on a graph chart or map.

MACRO–COMMAND

A group of a series of commands redefined as a single command.

MATRIX

A table which has a regular rows–by–columns structure in which the rows represent elements of a larger category and, similarly, the columns represent elements of another larger category. The data in the cells of the matrix are the values of the condition specified by the row element and the column element. A spreadsheet and a correlation matrix are representative examples of a matrix.

MAIN MENU

A top–level menu displayed upon entry into the system.

MENU (MENU SELECTION)

A type of dialogue in which a user selects one item out of a list of displayed alternatives. Selection may be made by pointing and clicking, associated option code, or by an adjacent function key.

MENU BAR

A specialized fiinction area that displays categories of user response altematives.

MENU PALETTE

A special kind of user–requested menu that provides a matrix of cells containing various items, for example, graphics or painting "tools." While the format is different, a menu palette functions in the same way as a standard pull–down or pop–up menu.

MESSAGE AREA

A specialized function area for text communication from another user at a different workstation or delivered automatically by the system to describe a system state or operation (e.g., a status message).

MONITOR

The physical device housing the electronics, display, and display controls for an interactive computer system.

MULTIPURPOSE APPLICATION CONSOLE (MPAC)

The electronic core workstation including displays, keyboard, and controllers.

MULTITASKING

The parallel performance of two or more tasks.

NATURAL LANGUAGE

A means of interacting with computers whereby people use a familiar natural language (such as English) to operate and give instructions. Users do not have to learn a command syntax nor select from menus.

NONDISRUPTIVE

An action that does not interfere with the user's ongoing activities.

OBJECTS

A distinct information unit whose representation can be displayed and/or manipulated by the information system. Objects are normally represented by graphic icons and/or names.

OPEN WINDOW (See WINDOW.)

Windows which are both perceptually and functionally available to the user. Two types of open windows exist, active and inactive.

OVERSTRIKE MODE

A data entry mode which allows the user to type a new character by entry directly over the old one. The original characters are lost as new ones are typed.

PAGING

A method of viewing and moving through data in which a user conceives of data as being grouped into display sized pages and moves through it by discrete steps.

PARENT MENU

The application level menu displayed upon entry into an application.

PASTE

A system function that puts the contents of the temporary editing buffer (a selection previously cut or copied) at the insertion point of the current interactive window. The buffer contents are not altered by this operation.

PERMANENT MENU

Menus which are constantly visible and [illegible] separable part of the display. These menus cannot be removed or hidden away [illegible] the entire window is closed or the display itself changes.

PLACEHOLDING CURSOR

See CURSOR.

POINTING CURSOR

See CURSOR.

POP–UP MENU

Similar in appearance and function to pull–down menus, they are activated or brought into full view by a complete selection action: for example, the pressing and releasing of a selection button. Menu items are selected by a selection action on the desired menu entry. Pop–up menus remain visible until another user action takes place to hide the menu or make a selection. If the user wants to hide the menu without making a selection, there is generally a close box or "exit menu" item available.

PRINT QUEUE

An area of computer memory that temporarily stores a file to be printed so that the user can continue interacting with the system while the file prints.

PROGRAMMABLE FUNCTION KEYS

User–programmable keys whose function may vary between applications or between users within an application.

PROTOTYPE

A training/evaluation model of an interface which includes the functions and capabilities expected in the final system, though not in a finished form.

PULL–DOWN MENU

A menu whose items are normally "hidden" from the user's view and accessed by the user holding the selection button down over the desired menu bar label. Selection of the text label activates the presentation of a list of menu items which are attached to the menu bar giving the user the impression that the list of items was pulled down from the menu bar. While the selection button is down, the user can move the cursor over the selections and release the selection button over the desired menu item. This menu is only visible to the user as long as the selection button remains depressed.

RELEARNING TIME

A measure of the amount of work the user must accomplish in order to achieve a previous level of competence on the system.

RETRIEVAL BUFFER

Permits the user to retrieve data after an action that would otherwise have destroyed the data (e.g., saving changes to a file and thereby destroying the old data in that file or deleting a file).

ROTATION OF AN OBJECT

The moving of an object around an imaginary line through the center of the object clockwise or counter–clockwise 90 degrees. The rotation movement is not constrained to the plane of the display.

SCREEN

The software controlled, visual interface of a monitor.

SCREEN DUMP

An action, usually performed with a keystroke sequence, that causes the exact contents of the current screen display to be captured for printing or storage in a file.

SCREEN STRUCTURE

A generic display element such as a menu bar or title.

SCROLLING

A method of viewing data in which the user conceives of data as having continuous vertical or horizontal movement within a set of linked displays (e.g., a text file) behind a fixed display window.

SCROLLING/PAGING STRUCTURE

a display structure for scrolling and paging which permits the user to move either horizontally or vertically through a display or connected sequence of displays. Scrolling provides the appearance of continuous movement, whereas paging provides movement in discrete steps. (See also PAGING, SCROLLING.)

SELECTING

A user's action of identifying display elements to the computer in order to ready them for use in some way (e.g., to move them, change their attribute(s), or delete them, usually accomplished with an input device click).

SINGLE–ACTION

A functional collection of discrete actions resulting in a clearly specified action (e.g., selection of a menu item or the entry of a command key sequence).

SPEECH RECOGNITION

Permits a user to provide spoken input which a computer interprets as data or commands.

STROKE WIDTH

The width of a line comprising a character.

SYSTEM RESPONSE TIME

The elapsed time between the initiation of a command and the notification to the user that the command has been completed.

SYSTEM STATUS INFORMATION

Current data processing information which is displayed to a user either automatically or by user request, perhaps indicating system load, keyboard lock, or processing delay.

TABLE

A rows and columns structure consisting of functional areas which contain data which may or may not require any input. Tables may be used to present a variety of types of information.

TASK ANALYSIS

A method of detailing the components of a task in terms of the demands placed upon the human operator, the information required by the operator, the extent to which the task requires reliance on or coordination with other personnel, and the relation of the task to other tasks.

TEMPORARY EDITING BUFFER

(Known on some systems as the clipboard) Is normally invisible to the user, but may be displayed in a window. This buffer is independent of, but able to interface with, all other applications. Its purpose is to hold data temporarily so that it can be moved from one place in a file to another or from one file to another.

TEXT

The primary display for word processing, consists of alphanumeric character strings in linear arrays, making up words, sentences, and paragraphs.

TILING

A means of manipulating windows by which multiple windows on the same display abut but do not overlap. As the number of windows increases in the tiled window environment, the size of each window decreases.

TITLE

A unique identifier, distinguishable from other screen structures, which describes a display.

TOUCH ZONE

An area of a display, visible to the user, that a user can activate to perform a predefined operation (e.g., displaying a pop–up window).

TRANSFER OF TRAINING

To the extent that the user trained on the first interface learns the second one more quickly/is more accurate/makes fewer procedural errors than a novice trained solely on the second interface, positive transfer of training has occurred. If training resulted in longer training time or less proficient performance with a novel application, negative transfer of training has occurred.

UNDO

A capability that reverses the effect of the previous operation.

USER DEFINABLE DIALOGUE COMPONENTS

Allow users to assign a single component of the dialogue (e.g., a term in the UIL or a key in a set of function keys) to a single command or to a series of commands. The user can thereafter use that dialogue component to elicit those commands.

USER GUIDANCE

Computer prompts and feedback that aid the user in performing a task. (See SCREEN STRUCTURE.)

USER INTERFACE LANGUAGE (UIL)

English–like word strings in a syntactic structure that may be keyed (or potentially, spoken) onto a command line.

USER RESPONSE TIME

The speed with which a user can enter commands and control a system regardless of the computer's ability to quickly process the commands.

USER SUPPORT ENVIRONMENT (USE)

The common standardized interface supporting effective and efficient communication between users and computer systems in the performance of many computerized functions employed in the Space Station Freedom Program.

VISUAL ANGLE

A measure (in degrees) of the size of the retinal image subtended by some displayed object.

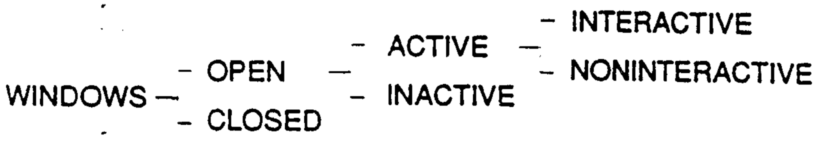
WINDOW

A rectangular, visually distinct portion of a display screen showing a particular type of information or function. Multiple windows may exist. Each is an independent display element moveable, scrollable, and adjustable in size.

Open/Closed — When a window is opened, it appears on the screen. Windows may be closed (removed from the screen) and reopened.

Active/Inactive — Open windows may be active, with some on–going process occurring, or inactive.

Interactive/Noninteractive — Active windows may be interactive (receptive to user input) or noninteractive.



WORD WRAP

Occurs when words displaced from one line are moved to the next line so as to maintain the continuity of the text.

X–Y CONTROLLERS

Controllers which have the ability to control the cursor or other followers in the x and y dimensions.

X–Y–Z CONTROLLERS

Controllers which have the ability to control the cursor or other followers in the x and y dimensions and the screen and to provide control of apparent movement in the z dimension.

APPENDIX C TASK ANALYSIS TABLES

TABLE 1 – POWER MANAGEMENT AND DISTRIBUTION (PMAD) TASK ANALYSIS

INPUTS refer to an active computer–the computer signals displays, or otherwise presents information to the user or the information the user carries in memory from a recent operation (calculation of a figure, extraction from a display, etc.). The TASK describes what the user does or is to do. OUTPUTS refer to the overt actions of the user as regards the computer or the information obtained or decision made as a result of a cognitive process. REQUIREMENTS provide greater detail regarding the actions, decision aids, computer help, etc. that facilitate the provision of inputs to the user or outputs to the user or computer.

INPUTS	TASK	OUTPUTS
1. start–up display	activate PMAD sub–program	PMAD start–up display. REQUIREMENT: a method of selection or indicating the sub–program required (menu, command language, etc.).
2. PMAD start–up display	display PMAD menu	choose menu display option. REQUIREMENT: method of selecting menu OR *automatic presentation* of menu when PMAD sub–program is activated.
3. PMAD menu. REQUIREMENT display of relevant options.	select Power Management Control (PMC) from main menu.	PMC display. Selected from the four possible sub–systems in PMAD. REQUIREMENT: method of selection (command language, highlighting, double clicking, etc.).
4. PMC display. (includes all functions related to PMC).	search for level of energy in storage cells.	determination of stored power. REQUIREMENT: display of available power in a manner consistent with the level of precision required for proper completion of the task (analog for less precision, *digital* for greater precision).
5. PMC display. stored power values	compare optimal with actual values	determination of optimal values, note relative similarities or dissimilarities. REQUIREMENT: ready availability of optimal values allocated either to the computer, the user's memory, hardcopy, or some help/reference function.

6. PMC display. actual and optimal energy storage levels. REQUIREMENT: (blinking, highlighting, method of keeping both values in the memory of user.	evaluate storage system status	determination of system status. REQUIREMENT: decision rule or computer generated decision aid computer message, etc.).
7. PMC display. knowledge of system status.	scan tank temperatures (H.0,H20) periodically	determination of tank temperatures. Assumed are the tasks recognize and evaluate to ensure that tank temperatures aren't out of range. REQUIREMENT: display of tank temperatures and some indication of in range/out of range.
8. PMC display. system status (storage status, tank temperature, production level).	scan energy production rate.	determination of production level from PMAD (may or may not be necessary to determine the production of individual SD and PV elements). REQUIREMENT: presentation of or ability to determine production rate (either through analog or digital display).
9. PMC display. syctem status knowledge (storage status, tank temperature, production level)	display short term plan (STP).	STP display REQUIREMENT: STP must be integrated within (chosen from) the PMC (and possibly all sub–system) display(s).
10. STP display. system status knowledge (storage status, tank temperature, production levels):	search for anticipated production rate.	determination of production rate as anticipated in the STP.
11. STP display (storage status, tank temperatures, production level, anticipated production level)	compare actual with anticipated production rate (differential production figure.)	notation of anticipated versus actual production
12. STP display (storage status, tank temperatures, differential I environmental production figure)	evaluate production rate relative to configuration to meet circumstances.	determination of necessity of changing system anticipated energy storage and production in light of current storage and production. REQUIREMENT: necessary for the operator to remember or have differential production figure(s).

<< Decision is made to re–configure power system >>

13. STP display	exit STP	PMC display. REQUIREMENT: exit/close/end commands in the form of command language, function areas, menu selection, etc.
13a. PMC display	exit PMC display	PMAD main menu. REQUIREMENT: exit/close/end commands in the form of command language, function areas, menu selection, etc. (see also step 13).
14. PMAD main menu	select Power Source Control (PSC) from PMAD main menu.	PSC display. Choice of PSC from the four possible sub–systems on the menu (see also step 3).
15. PSC display. differential production figure.	formulate response to adjust production rate.	determination of steps necessary to adjust production rate. REQUIREMENT: operator knowledge of or access to (through the PMAD displays) the steps available for changing the production rate. May be incorporated into a separate menu or help function, or as an integrated part of the display/control screen.
16. PSC display differential production figure, system response.	project rotation of solar array(s) to adjust production rate.	new configuration coordinates for the solar array(s). REQUIREMENT: a method for determining change in angles, possibilities include digital or graphic displays, or video monitors.
17 PSC display. differential production figure, anticipated new production rate, new solar array coordinates.	align solar wing visually/ cognitively. graphic displays, or video monitors (see also step 16).	REQUIREMENT: a method for determining change in angles; possibilities include digital or
18. PSC display. differential production figure, anticipated new production rate, new solar array coordinates.	adjust array	activating the system by a "switch"– either a manual, but probably a computer, switch that activates adjustment mechanisms to orient the array to coordinates determined in step 16. REQUIREMENT: interaction between coordinates and adjustment mechanisms, a specified section on the display for entering in new configuration.
19. PSC display. differential production figure.	examine gimbal status of mirror.	Current gimbal coordinates. REQUIREMENT: display of mirror gimbal status.
20. PSC display differential production figure, current gimbal status.	project rotation of mirror to adjust production rate.	new configuration coordinates for the solar mirror(s). REQUIREMENT: a method for determining change in angles; possibilities include digital or graphic displays, or video monitors.

21. PSC display differential production figure, current gimbal status, new gimbal coordinates, anticipated production rate.	adjust mirror	activating the system by a "switch"– either a manual, but probably a computer, switch that activates adjustment mechanisms to orient the array to coordinates determined in step 20. REQUIREMENT: interaction between coordinates and adjustment mechanisms, a specified section on the display for entering in new configuration (see also step 18).
22. PSC display. new gimbal configuration, differential production figure, current production	scan PV and SD readouts REQUIREMENT: PV and SD rate.	determination of effectiveness of change in relevant PV and SD readouts (delta storage level). readouts should be displayed such that rate of change information can be easily obtained; possibilities include digital or graphic displays.
23. PSC display. delta storage level.	detect values out of range	determination of value range status. REQUIREMENT: clear knowledge of what constitutes an out of range variable allocated either to the system, user, or hardcopy reference.
24. PSC display. system status, delta storage level.	search for energy level in storage cells	Current energy level in storage. View ESS health status noting the effectiveness of the change in configuration. REQUIREMENT: knowledge of optimal ESS figures–allocated either to the system, user, or hardcopy.
25. PSC display. baseline energy storage level. REQUIREMENT: remember or find from display baseline energy storage level.	compare current with baseline level	determination of rate of change (see steps 8 and 23).
26. PSC display. rate of change	examine production rate	determination of the acceptability of the new production rate.
27. PSC display. production rate, baseline energy storage level.	formulate duration of solar energy collection	estimate, using previous actual storage values and current production rate, of the duration of solar energy collection at current configuration.
28. PSC display. duration of production, production rate, baseline energy storage level.	display STP	STP display. REQUIREMENT: STP should be integrated within/chosen from the PMC (and possibly all sub–system) display(s).
29. STP. duration of production, production rate, baseline energy storage level.	formulate new values for relevant entries.	new entries. REQUIREMENT: method of identifying the values that need to be changed and calculating the new values.

30. STP. new values for schedule	enter new values	updated STP. REQUIREMENT: method of entering new values (keyboard, mouse, speech) and checking the new entries for accuracy.
31. STP	exit STP	PSC display REQUIREMENT: see step 13.
32. PSC display	exit PSC	PMAD main menu REQUIREMENT: see step 13a.
33. PMAD main menu	select main bus switch control (MBSC)	MBSC display (see steps 3 and 14).
34. MBSC display	scan values for primary distribution cables and switchgear	determination of cable and switchgear values. REQUIREMENT: display of cable and switchgear values such that they are easily scanned.
35. MBSC display	detect values out of range	determination of value range status. REQUIREMENT: clear knowledge of what constitutes an out of range variable allocated to the system, user, or hardcopy.
36. MBSC display. out of range values.	formulate new settings for systems with values out of range.	new values. REQUIREMENT: method for calculating new values–allocated to the system, user, or hardcopy.
37. MBSC display. out of range values, new values.	adjust values as necessary	new entries for relevant systems. REQUIREMENT: method for entering new values (keyboard, mouse, speech) and checking the new entries for accuracy) (see also step 30).
38. MBSC display. new values.	scan values	notation of new values.
39. MBSC display. new values, optimal values.	compare newly adjusted with optimal values	determination of effectiveness of adjustment. REQUIREMENT: comparison of newly adjusted with optimal values–directly or in terms of rate of change.
40. MBSC display.	evaluate system status	decision regarding system status–either go to next step or reiterate from step 34. REQUIREMENT: method of determining or interpreting values indicating acceptable system status.
41. MBSC display.	enter acknowledgement of of system status	computer acknowledges acknowledgement. Acknowledgement may or may not be desirable for all system status changes. If acknowledgement is necessary, use visual, auditory, or both.

42. MBSC display. acknowledgement OK'ed.	exit MBSC	PMAD main menu (see steps 13 and 32).
43. PMAD main menu	select Power Distribution Control (PDC)	selection of PDC from main menu (see steps 3, 14, and 33).
44. PDC display	scan power drain/usage at usage centers.	general determination of overall and sub–system drains. REQUIREMENT: display of current usage and possibly anticipated usage, or some difference figure.
45. PDC display. system usage.	search usage at particular centers	detailed knowledge of usage at particular centers warranting closer examination. REQUIREMENT: ability to extract more detailed information, either information already displayed on the screen or contained within a window related to the usage center in question.
46. PDC display. current usage, anticipated usage, related information.	examine power usage REQUIREMENT: method of noting findings (hardcopy, clipboard, comments area, etc.).	notation of causes of findings.
47. PDC display. cause of discrepancy.	evaluate usage status	determination of status and decision to continue to next step or to reiterate from step 45.
<< discrepancy resolved, decision to continue >>		
48. PDC display	display STP	STP. REQUIREMENT: STP should be integrated within/chosen from the PMC (and possibly all sub–system) displays.
49. STP. current power drain, anticipated drain.	compare current and anticipated usage with STP estimate.	determination of power drain status. REQUIREMENT: method of remembering relevant values from PDC so as to compare them with the STP.
50. STP. current drain, anticipated drain.	formulate new values as STP entries	calculation of STP entries.
51. STP new STP entries	enter values	updated STP. REQUIREMENT: method of entering new values (keyboard, mouse, speech) and checking the new entries for accuracy.
52. STP	exit STP,	PDC display
53. PDC display	exit PDC display	PMAD main menu
54. PMAD main menu	exit PMAD main menu	SSIS display

TABLE 2 – REBOOST OPERATION TASK DESCRIPTION

Reboost Operation: one shift's work with reboost as detailed by section 5.1 of LEMSCO–23592 (A Definition of the Operations Management System (OMS) Relationships).

communicate	enter logon id	1
	receive briefing	2
	read shift report	3
check	evaluate system status (ECLS, C&W, TCS, PROP, EPS, GN&C, DMS, C&T, MSC)	4
command	direct preparation for reboost	5
	activate preparation of propulsion system	6
communicate	enter thrust/thruster characteristics	7
plan	formulate alternative times for reboost	8
	formulate alternative delta velocities	9
check	evaluate the criticality of performing maneuver	10
communicate	enter annunciation parameters	11
check	compare optimal reboost ignition time with short term plan	12
plan	formulate new short term plan to include reboost	13
data management	display short term plan	14
communication	enter new short–term plan	15
check	examine the validity of the integrated reboost procedures	16
check	examine the redundancy management required to support reboost	17
	compare time selected against requirement and criticality needs	18
	evaluate severity of conflicts	19
plan	select schedule or reschedule	20
check	compare impact of reboost against relevant core systems, payload activities, and resource requirements	21
monitor	search for beam vector	22
command	direct information integration into data flow	23
data management	mail updated plans and reallocated resources to OMGA and mission planners	24
command	activate transmission inhibits	25
	activate interlock reboost status	26
	stow loose items within Space Station Freedom	27
check	examine Space Station Freedom exterior for plume impingement	28
spatially orient	locate a safe path	29
	stow tools	30

command	activate CRT timer	31
plan	project safe back out procedures for each reboost step	32
command	activate heaters and temperature monitors	33
plan	formulate station mass properties	34
communicate	receive a state vector update from ground as back up	35
data management	save back up	36
communicate	enter new Kalman filter constants	37
	enter new burn pad coordinates	38
command	activate dedicated displays	39
check	examine state vector for quality assurance	40
communicate	enter quality assurance results on OMS	41
data management	save results	42
command	activate targeting tasks	43
compar	examine target options	44
plan	select an option	45
compare	evaluate target solution	46
command	direct all traffic out of control zone	47
communicate	transmit alert to all users	48
	enter target set to be applied	49
check	evaluate feed system health	50
monitor	search displays	51
	extract quantity values	52
check	compare actual with specified quantities	53
monitor	detect out–of–range values	54
command	activate fluid loading	55
	actuate fluid pressurization	56
plan	select tank	57
monitor	examine pressure regulator	58
check	compare actual with optimal pressures	59
monitor	detect out–of–range values	60
command	activate valve configuration sequentially	61
monitor	search for flow rate at sink and source	62
check	compare actual with specified values	63
monitor	detect–out–of–range values	64
command	activate fluid transfer to sink	65
plan	formulate quantity of fluid leaked	66
communicate	enter amount of leakage	67
data management	save data	68

check	evaluate engine health (OMS std test)	69
plan	project delta velocity available	70
	formulate propellant paths available	71
	select a propellant path	72
	formulate downmode options	73
check	evaluate downmode options	74
command	direct burn simulation preparation	75
plan	select thrust source	76
	select bask up options	77
	select time for simulation	78
command	direct systems to engage simulation modes	79
data management	display procedures from appropriate libraries	80
plan	read procedures	81
command	direct loading of predetermined back out procedures onto system	82
	actuate simulation configuration	83
communicate	transmit expected thruster performance to GN&C	84
command	activate simulation	85
monitor	scan displays	86
check	compare actual with specified instrument values	87
monitor	detect out–of–range values	88
command	direct system to remain within tolerance ranges	89
	deactivate simulation	90
check	evaluate simulated burn data	91
command	direct return to support operations configuration	92
data management	display results of simulation	93
check	compare solution attained with target solution	94
communicate	transmit to OMS simulation completion	95
	enter activities and results in log	96
data management	save log	97
plan	select source reallocation as required	98
command	activate hangar door closing	99
communicate	inform crew of imminent maneuver	100
	enter rate as commanded	101
plan	formulate burn attitude	102
command	activate maneuver to burn attitude	103
communicate	enter completion of maneuver in log	104
check	evaluate system readiness	105
	examine reboost switch checklist	106
	examine valve checklist	107

command	direct CMG disengagement	108
	activate RCS attitude control circuitry	109
	—mode CMG's for desaturation— ??	110
plan	formulate required thrust vector offset to desaturate CMG's	111
communicate	enter propulsion system adjustments to offset thrust vector	112
	enter transition to RCS control in log	113
command	deactivate star trackers	114
communicate	enter deactivation of star trackers in log	115
	enter C&W thresholds	116
check	evaluate control zone clearance	117
	evaluate platform stabilization and clearance	118
command	deactivate preparatory heaters	119
	—apply additional load shielding as required— ??	120
check	evaluate completion of prerequisite commands	121
command	activate burn	122
monitor	scan monitors	123
check	compare actual with specified values	124
monitor	detect out–of–range values	125
command	adjust attitude control mode to support coast phase	126
	activate line purge	127
	activate thruster blowdown	128
communicate	enter newly accumulated hours of operation	129
	enter completion of first burn in log	130
	enter burn results	131
command	direct return of systems to on–orbit support configuration	132

APPENDIX D REFERENCES

1. Anderson, B. L. Differences in teachers' judgement policies for varying numbers of verbal and numerical cues. Organizational Behavior and Human Performance, 1977, 19, 68–88.

2. Author. Human Engineering Design Criteria for Military Systems, Equipment and Facilities, MIL–STD–1472C, May 2, 1981.

3. Author. Human Engineering Guidelines for the Evaluation and Assessment of Video Display Units, NUEREG/CR–4227, 1985.

4. Author. Human Engineering Procedures Guide, DOD HDBK 763, 27 February 1987, Section 4.0.

5. Author. Space Station Information System User Interface Scope, NASA–JSC–32004, Houston, TX, 1986.

6. Author. Space Station Programs Information System Concept Document. Draft, July 7,1988.

7. Banks, W. P., Bodinger, D., and Illige, M. Visual detection accuracy and target noise proximity. Bulletin of the Psychonomic Society, 1974, 2, 411–414.

8. Beck, J. Effect of orientation and of shape similarity on perceptual grouping. Perception & Psychophysics, 1966, 1, 300–302.

9. Beirschwale, J. Speech Versus Keying Commands in a Text–Editing Task. Unpublished thesis, Texas A & M University, 1987.

10. Bennett, J. L. Managing to meet usability requirements: Establishing and meeting software development goals. In J. Bennett, D. Case, J. Sandelin, and M. Smith (Eds.), Visual Display Terminals. Englewood Cliffs, New Jersey:

11. Bly, S. A. and Rosenberg, J. K. A comparison of tiled and overlapping windows. Human Factors in Computing Systems–CHI '86 Proceedings, 101–106, 1986.

12. Bruning. J. L., and Kintz, B. L. Computational Handbook of Statistics. Glenview, Illinois: Scott, Foresman and Company, 1977.

13. Cahill, M. C., and Carter, R. C. Color code size for searching displays of different density. Human Factors, 1976, 18(3), 273–279.

14. Cakir, A., Hart, D. J., and Stewart, T. F. M. Visual display terminals. Chichester: John Wiley & Sons, 1980.

15. Christ, R. E. Review and analysis of color coding research for visual displays. Human Factors, 1975, 25(6), 683–692.

16. Connolly, D. W., Spanier, G., and Champion, F. Colour display evaluation for air traffic control. In V. Hopkin, Colour displays in air traffic control. IEEE Conference on Displays for Man–Machine Systems, 1977, 46–49.

17. Cooper, M. Human Factor Aspects of Voice Input/Output. Speech Technology, 2.

18. Desaulniers, D. R., Gillan, D., Rudisill, M., and Burns, M. The computer based display of procedures. Third Annual User–System Interface Conference Proceedings, 1988, 217–224.

19. Dodson, D. W., and Shields, N. L. Development of user guidelines for ECAS display design (V.1), (Report No. NASA–CR–150877). Huntsville, Alabama: Essex, 1978.

20. Duchnicky, R. L., and Kolers, P. A. Readability and text scrolled on visual display terminals as a function of window size. Human Factors, 1983, 25(6), 683–692.

21. Elkerton, J., and Williges, R. C. Information retrieval strategies in a file–search environment. Human Factors, 1984, 26(2), 171–184.

22. Ericsson, K. A., and Simon, H. A. Verbal reports as data. Psychological Review, 1980, 87, 215–251.

23. Eriksen, C. W., and Hoffman, J. E. Temporal and spatial characteristics of selective encoding from visual displays. Perception and Psychophysics, 1972, 12, 201–204.

24. Estes, W. K. Interactions of signal and background variables in visual processing. Perception & Psychophysics, 1972, 12, 278–286.

25. Francik, E. P., and Kane, R. M. Optimizing visual search and cursor movement in pull–down menus. Proceedings of the Human Factors Society 31st Annual Meeting, 1987, 718–721.

26. Goldsmith, T. E., and Schvaneveldt, R. W. The role of integral displays in decision making. In Human Factors in Computer Systems. Gaithersburg MD: Association for Computing Machinery, 1981, 197–210.

27. Good M., Spine, T. M., Whiteside, J., and George, P. User–derived impact analysis as a tool for usability engineering. CHI'86 Conference Proceedings, 1986, 241–246.

28. Gould, J. D., and Lewis, C. Designing for usability: Key principles and what designers think. Communications of the ACM, 1985, Vol. 28, 300–311.

29. Green, P., and Wei–Haas, L. The rapid development of user interfaces: Experience with the Wizard of Oz method. Proceedings of the Human Factors Society 29th Annual Meeting, 1985, 470–474.

30. Halasz, F. G. Reflections on Note Cards: Seven issues for the next generation of hypermedia systems. Hypertext '87 Papers. 1987, 345–366.

31. Hammond, K. R. A theory based review on theory and research in judgement and decision making. (Tech. Report 260) Boulder: University of Colorado, Center for Research on Judgement and Policy, 1986.

32. Hutchins, E. L., Hollan, J. D., and Norman, D. A. Direct manipulation interfaces. In D. A. Norman and S. W. Draper (Eds.), User Centered System Design. Hillsdale, N.J.: L. Erlbaum, 1986, 87–124.

33. Kahneman, D., and Henik. A. Effects of visual grouping on immediate recall and selective attention. In S. Domic (Ed.), Attention and performance VI. Hillsdale, N.J.: L. Erlbaum, 1977.

34. Kahneman, D., and Henik, A. Perceptual organization and attention. In M. Kubovy and J. R. Pomerantz (Eds.), Perceptual organization. Hillsdale, N.J.: Erlbaum, 1981.

35. Keppel, G. Design and Analysis: A researcher's handbook. Englewood Cliffs, New Jersey: Prentice–Hall, 1982.

36. Kerkar, S. P., and Howell, W. C. The effect of information display format on multiple–cue judgement. (ONR Tech. Report 80–2) Houston, TX: Rice University, 1984.

37. Kloster, B. V. and Tischer, K. Man–machine interface design process. In B. Shackel (ed.), Human–Computer Interaction–INTERACT '84. Amsterdam: Elsevier Science, 1985.

38. Knox, R. E., and Hoffman, P. J. Effects of variation of profile format on intelligence and sociability judgements. Journal of Applied Psychology, 1962, 46, 14–20.

39. Kosslyn, S. M. Graphics and human information processing. Journal of the American Statistical Association, 1985, Vol: 80, 499–512.

40. Krohn, G. S. Flowcharts used for procedural instructions. Human Factors, 1983, 25, 573–581.

41. Kruk, R. S., and Muter, P. Reading of continuous text on video screens. Human Factors, 1984, 26(3), 339–345.

42. Lalomia, M. J., and Happ, A. J. The effective use of color for text on the IBM 5153 color display. Proceedings of the Human Factors Society 31st Annual Meeting, 1987, 1091–1095.

43. Lea, W. L. The Value of Speech Recognition Systems. In Trends in Speech Recognition. 3–18.

44. Lenorovitz, D. R., Phillips, M. D., Ardrey, R. S., and Kloster, G. V. A taxonomic approach to characterizing human–computer interfaces. In G. Salvendy (ed.) Human–Computer Interaction. Amsterdam: Elsevier Science, 1984.

45. Lewis, R., Gillan, D. J., and Rudisill, M. Perceptual and informational dimensions of graphics. Unpublished manuscript, 1988.

46. Marchionini, G., and Shneiderman, B. Finding facts versus browsing knowledge in hypertext systems. Computer, Jan. 1988, 70–79.

47. McCauley, M. E. Human factors in voice technology. In F. A. Muckler (ed.), Human Factors Review: 1984. Santa Monica, CA: The Human Factors Society, 1984.

48. Muratore, D. A. User interface language operations concepts—Preliminary, NASA–JSC–22360, Houston, TX: 1986.

49. Murray, J. T., Praag, J., and Gilfoil, D. Voice vs. keyboard control of cursor motion. Proceedings of Human Factors Society Meeting. Santa Monica, California, 1983, 103.

50. Myers, B. A. Window interfaces: A taxonomy of window manager user interfaces. IEEE Computer Graphics and Applications. 1988, 65–83.

51. Navon, D., and Gopher, D. On the economy of the human processing system. Psychological Review, 1979, 86, 244–255.

52. North, R. Applications of Speech Recognition. In R. Pew, and P. Green (Eds.), Human Factors Engineering Short Course Notes. Ann Arbor, Michigan: The University of Michigan, Chrysler Center for Continuing Engineering Education, 1987.

53. Pace, B. J. Color combinations and contrast reversals on visual display units. Proceedings of the Human Factors Society, 28th Annual Meeting, 1984, 326–330.

54. Poock, G. K. Experiments with voice input for command and control, Technical Report NPS–55–80–016, Naval Postgraduate School, 1980.

55. Powers, M., Lashley, C., Sanchez, P., and Shneiderman, B. An experimental comparison of tabular and graphic data presentation. Journal of Man–Machine Studies, 1984, 20(6), 545–566.

56. Robinson, A., Sale, R., and Morrison, J. Elements of Cartography. New York: Wiley and Sons, 1978.

57. Robinson, C. P., and Eberts, R. E. Comparison of speech and pictorial Displays in a cockpit Environment. Human Factors, 1987, 29(1), 31–44.

58. Rogers, Y. Evaluating the meaningfulness of icon sets to represent command operations. Proceedings of HCI '86 Conference on People and Computers: Designing for Usability, 586–603. London: British Computer Society, 1986.

59. Schwartz, D. R., and Howell, W. C. Optional stopping performance under graphic and numeric CRT formatting. Human Factors, 1985, 27, 433–444.

60. Shneiderman, B. Designing the User Interface, Menlo Park, California, 1987.

61. Skriver, C. P. Vocal and Manual response modes: Comparison using a time–sharing paradigm, Report No. NADC–79127–60, Naval Air Development Center, 1979.

62. Smith, S. L., and Goodwin, N. C. Blink coding for information display. Human Factors, 1971, 13, 283–290.

63. Smith, S. L., and Goodwin, N. C. Another look at blinking displays. Human Factors, 1972, 14(4), 345–347.

64. Smith, S. L., and Mosier, J. N. Guidelines for Designing User Interface Software, ESD–TR–86–278, 1986.

65. Smith, V., Rushforth, S., Harris, D., and Moulden, B. Preliminary study of advantages to be gained by the use of colour coded information on electronic data and radar displays for air traffic control. In V. Hopkin, Colour displays in air traffic control. IEEE Conference on Displays for Man–Machine Systems, 1977, 46–49.

66. Spoehr, K. T., and Lehmkuhle, S. W. Visual Information Processing. San Francisco: W. H. Freeman, 1982.

67. Treisman, A. Perceptual grouping and attention in visual search for features and for objects. Journal of Experimental Psychology: Human Perception and Performance, 1982, 8(2), 194–214.

68. Treisman, A., and Gelade, G. A feature–integration theory of attention. Cognitive Psychology, 1980, 12, 97–136.

69. Treisman, A., Kahneman, D., and Burkell, J. Perceptual objects and the cost of filtering. Perception & Psychophysics, 1983, 33(6), 527–532.

70. Trollip, S. R., and Sales, G. Readability of computer–generated fill–justified text. Human Factors, 1986, 28(2), 159–164.

71. Tufte, E. R., The Visual Display of Quantitative Information. Cheshire, Connecticut: Graphics Press, 1983.

72. Tullis, T. S. An evaluation of alphanumeric, graphic and color information displays. Human Factors, 1981, 23, 541–550.

73. Tullis, T. S. A system for evaluating screen formats. Proceedings of the Human Factors Society 30th Annual Meeting, 1986, 1216–1220.

74. Tullis, T. S. Practical issues in designing direct manipulation screens. Unpublished manuscript, 1987.

75. Vartabedian, A. G. The effects of letter size, case, and generation method on CRT display search time. Human Factors, 1971, 13, 363–368.

76. Webb, Campbell, Schwartz, and Sechrest. Unobtrusive Measures: Nonreactive Research in the Social Sciences. Chicago: Rand McNally, 1966.

77. Welch, J. R. Automatic data entry analysis, Technical Report RADC–TR–77–306, Rome Air Development Center, 1977.

78. Wickens, C. D. The structure of processing resources. In R. Nickerson and R. Pew (eds.), Attention and Performance VIII. New York: Erlbaum, 1980.

79. Wickens, C. D. Engineering psychology and human performance. Columbus: Merrill, 1984.

80. Wickens, C. D., Sandry, D. L., and Vidulich, M. Compatibility and response competition between modalities of input, central processing and output. Human Factors, 1983, 25, 227–248.

81. Williges, B. H., Schurick, J. M., Spine, T. M., and Hakkinen, M. T. Using speech in the human–computer interface. In R. W. Ehrich and R. C. Williges (eds.), Human–Computer Dialogue Design. Amsterdam: Elsevier, 1986.

82. Woodson, W. E. Visual Displays. Human Factors Design Handbook. New York: McGraw–Hill Book Company, 1982.

APPENDIX E INDEX

PRECEDING PAGE BLANK NOT FILMED

SPACE STATION FREEDOM PROGRAM OFFICE
HUMAN COMPUTER INTERFACE GUIDE

LIST OF CHANGES
JULY 1993

All changes to paragraphs, tables, and figures in this document are shown below:

SSCBD	ENTRY DATE	CHANGE	PARAGRAPHS
BB003118	06–15–93	REVISION A	2.1, 3.1, 3.1.1, 3.2.9, 3.2.10, 3.2.13, 3.2.16, 3.4, 6.1, 6.1.1, 6.1.2, 6.2.2, and 6.3.2
			APPENDIXES
BB003118	06–15–93	REVISION A	A and E
			PARAGRAPHS
BB003681	06–15–93	REVISION A	ALL