

U.S. Marine Corps



**INFORMATION RESOURCES
MANAGEMENT (IRM)
HUMAN COMPUTER INTERFACE
(HCI)
STYLE GUIDE**



UNITED STATES MARINE CORPS
MARINE CORPS COMPUTER AND TELECOMMUNICATIONS ACTIVITY
3255 MYERS AVENUE
QUANTICO, VIRGINIA 22134-5048

IN REPLY REFER TO:
5234/09
CTAS-40
15 JUL 19

From: Director, Marine Corps Computer and Telecommunications Activity

Subj: INFORMATION RESOURCES MANAGEMENT (IRM) HUMAN COMPUTER INTERFACE (HCI) STYLE GUIDE

Ref: (a) MCO 1510.37A, ITS System for Data System OccField 40
(b) MCO P5231.1B, Life Cycle Management of Automated Information Systems Projects (LCM-AIS)
(c) MCO 5271.1A, Information Resources Management (IRM) Standards and Guidelines Program

Encl: (1) DoD Human Computer Interface (HCI) Style Guide, Version 2.0, 30 SEP 92

1. PURPOSE. To provide a common framework for Human Computer Interface design and implementation as required by references (a) and (b). The enclosure is intended as a precursor to forthcoming standards to be issued by DoD in the area of HCI.

2. CANCELLATION. IRM-5234-02 of 17 MAR 1987.

3. AUTHORITY. This publication is distributed under the auspices of reference (c).

4. APPLICABILITY. The guidance contained in this publication is applicable to all Marine Corps personnel and contractors responsible for the design of applications requiring the use of HCI.

5. DISTRIBUTION. This technical publication will be distributed as indicated.

6. SCOPE

a. Compliance. Compliance with the provisions of this publication is required unless a specific waiver is authorized.

b. Waivers. Waivers to the provisions of this publication will be authorized only by CMC (MCCTA) on a case by case basis.

7. RECOMMENDATIONS. Recommendations concerning the contents of the enclosure are strongly encouraged and should be submitted to Director, MCCTA, at the following address:

5234/09
CTAS-40

Subj: INFORMATION RESOURCES MANAGEMENT (IRM) HUMAN COMPUTER
INTERFACE (HCI) STYLE GUIDE

United States Marine Corps
MARCORCOMTELECT
(CTAS)
3255 Myers Ave
Quantico, VA 22134-5048

MARCORCOMTELECT QUANTICO VA OMB (GICIZZ:MQGMCCTA)

8. SPONSOR. The sponsor of this technical publication is CMC
(MCCTA).



D. P. HOUSTON
Colonel, U.S. Marine Corps
Director, Marine Corps
Computer and
Telecommunications Activity

DISTRIBUTION STATEMENT "A"

DISTRIBUTION: PCN 186 523409 00

Copy to: 8145001

UNITED STATES MARINE CORPS

Information Resources Management (IRM) Standards
and Guidelines Program

Human Computer Interface (HCI) Style Guide
IRM-5234-09

Enclosure (1)

Human Computer Interface (HCI) Style Guide
IRM-5234-09

(This page intentionally left blank)

Human Computer Interface (HCI) Style Guide
IRM-5234-09

TECHNICAL PUBLICATION LIBRARY MAINTENANCE

The Information Resources Management Standards and Guidelines Program publications will be maintained at each receiving activity. Each activity is responsible for ensuring that their set of technical publications is complete, and that all published changes are promptly incorporated.

RECORD OF CHANGES

Change Number	Date of Change	Date Received	Date Entered	Signature of Person Entering Change

Human Computer Interface (HCI) Style Guide
IRM-5234-09

(This page intentionally left blank)

DEFENSE INFORMATION SYSTEMS AGENCY

Center for Information Management

Human Computer Interface

Style Guide



Version 2.0

September 30, 1992

ACKNOWLEDGEMENTS

DoD HCI Style Guide, Version 1:

The US Department of Defense Intelligence Information Systems (DODIIS) Style Guide was the foundation for the main body and Appendices A and B. The DODIIS Style Guide established the initial step of Human Computer Interface (HCI) standardization by providing the framework for future applications. The DODIIS Style Guide was produced by an initiative of the Defense Intelligence Agency with assistance from the MITRE Corporation, Center for Integrated Intelligence Systems.

Appendix C of the DoD HCI Style Guide was derived from the Human Factors Design Guidelines for the Army Tactical Command and Control (ATCCS) Soldier Machine Interface, Version 1. The Guidelines document was prepared for the Commander, ATCCS Experimentation Site (AES), Fort Lewis, Washington, 98433, in accordance with the provisions of ADEA/AES SOW #300A13442 and under a Related Services Agreement with the U.S. Department of Energy Contract DE-AC06-76RLO 1830. The Guidelines document was prepared by Pacific Northwest Laboratory (PNL), a National Laboratory operated for the U.S. Department of Energy by Battelle Memorial Institute.

DoD HCI Style Guide, Version 2:

Version 2 was created by combining Version 1 of the DoD HCI Style Guide with the Human Factors Design Guidelines for the Army Tactical Command and Control (ATCCS) Soldier Machine Interface, Version 2. The document was prepared by Pacific Northwest Laboratory (PNL), a National Laboratory operated for the U.S. Department of Energy by Battelle Memorial Institute.

TABLE OF CONTENTS

1.0	INTRODUCTION	1-1
1.1	BACKGROUND	1-1
1.2	PURPOSE	1-2
1.3	BASELINE	1-2
1.4	HUMAN-COMPUTER INTERFACE	1-4
1.5	SCOPE	1-5
1.6	INTENDED AUDIENCE	1-5
1.7	DESIGN GOALS	1-6
1.8	ASSUMPTIONS	1-7
1.9	STYLE GUIDE ORGANIZATION	1-8
2.0	MOTIF AND OPEN LOOK STYLISTIC DIFFERENCES	2-1
2.1	DIFFERENCES IN TERMINOLOGY	2-1
2.2	DIFFERENCES IN APPEARANCE	2-3
2.3	DIFFERENCES IN FEEL	2-4
	2.3.1 Mouse Button Definition Differences	2-4
	2.3.2 Keyboard Usage Differences	2-6
	2.3.3 Differences in the Use of Controls	2-7
2.4	STYLE GUIDE PRESENTATION	2-9
2.5	CONCLUSION	2-9
3.0	HARDWARE	3-1
3.1	INPUT DEVICES AND PROCEDURES	3-2
	3.1.1 Pointing Devices	3-2
	3.1.2 The Keyboard	3-4
	3.1.3 Input Focus	3-5
3.2	SPECIAL DISPLAYS	3-5
	3.2.1 Flat Panel Technology	3-6
	3.2.2 Liquid Crystal Displays (LCDs)	3-10
	3.2.3 Gas Plasma Displays	3-11
	3.2.4 Electroluminescence (EL) Displays	3-12
	3.2.5 Glare-Reducing Techniques	3-14
	3.2.6 Large-Screen Displays	3-16
	3.2.7 Stereoscopic/3D Displays	3-20
	3.2.8 Touch Interactive Devices	3-23
3.3	ALTERNATE INPUT/OUTPUT DEVICES	3-24
	3.3.1 Visual I/O	3-26
	3.3.2 Hearing I/O	3-27
	3.3.3 Mobility	3-27

TABLE OF CONTENTS (Cont.)

4.0	SCREEN DESIGN	4-1
4.1	INITIAL SCREEN DESIGN	4-2
4.1.1	Workstation Log-In	4-2
4.1.2	Application Log-In	4-4
4.1.3	Application Log-Off	4-4
4.1.4	Workstation Log-Off	4-4
4.1.5	Initial Workstation Screen Display	4-4
4.2	SCREEN DESIGN GUIDELINES	4-6
4.2.1	General	4-6
4.2.2	Consistency	4-6
4.2.3	Format	4-8
4.2.4	Grouping	4-14
4.2.5	Multiple Screens	4-15
4.3	COLOR	4-15
4.3.1	General	4-18
4.3.2	Color Selection	4-20
4.3.3	Tonal Color Coding	4-25
4.3.4	Color-Coded Symbols	4-25
4.3.5	Map Graphics and Color	4-26
5.0	WINDOWS	5-1
5.1	WINDOW BASICS	5-4
5.1.1	Basic Window Appearance	5-4
5.1.2	Dragging the Window	5-6
5.1.3	Scroll Bars	5-6
5.1.4	Application Area	5-6
5.1.5	Message Area	5-6
5.1.6	Resizing the Window	5-7
5.1.7	Window Controls	5-8
5.1.8	Window Colors/Patterns/Audio Signals	5-8
5.2	WINDOW DESIGN	5-10
5.2.1	General Guidance	5-10
5.2.2	Window Control	5-13
5.2.3	Designation	5-20
5.2.4	Labeling	5-21
5.2.5	Open Window Navigation	5-21

TABLE OF CONTENTS (Cont.)

6.0	MENU DESIGN	6-1
6.1	GENERAL	6-2
6.1.1	Consider Response Time and Display Rate	6-2
6.1.2	Instructions and Error Messages	6-2
6.1.3	Explicit Option Display	6-3
6.1.4	Stacking Menu Selections	6-3
6.1.5	Menus Distinct from Other Displayed Information	6-3
6.1.6	Menu Bars	6-3
6.1.7	Pull-Down Menus	6-4
6.1.8	Pop-Up Menus	6-5
6.2	FORMAT	6-6
6.2.1	General	6-6
6.3	HIERARCHICAL MENUS	6-8
6.3.1	Usage	6-8
6.3.2	General Guidance	6-8
6.3.3	Navigating Hierarchical Menus	6-8
6.3.4	Hierarchical Menu Tree Depth and Breadth	6-11
6.4	ITEM SELECTION	6-13
6.4.1	General	6-13
6.4.2	Selection by Pointing	6-14
6.5	MENU OPTION LABELING	6-15
6.5.1	General	6-15
6.5.2	Selector	6-16
6.6	DIALOG BOXES/POP-UP WINDOWS	6-17
6.6.1	Message Wording Guidelines	6-18
6.6.2	Work-In-Progress Window	6-19
6.6.3	Information Box	6-19
6.6.4	Caution/Warning Box	6-20
6.6.5	Menu Box	6-20
7.0	OBJECT ORIENTATION	7-1
7.1	DIRECT MANIPULATION	7-4
7.1.1	General	7-4
7.1.2	Metaphors	7-5
7.1.3	Icons	7-6
7.2	COMMAND ICONS	7-12
7.2.1	Command Icon Design	7-13
7.2.2	Command Icon Applications	7-14

TABLE OF CONTENTS (Cont.)

8.0	COMMON FEATURES	8-1
8.1	TACTICAL SYSTEM COMMON FEATURES	8-1
8.1.1	Date/Time Display	8-1
8.1.2	Latitude/Longitude Display	8-2
8.1.3	User-Definable Parameters	8-3
8.1.4	Wild Card Characters	8-4
8.2	ON-LINE HELP	8-4
8.2.1	Types of HELP	8-6
8.2.2	General Design	8-6
8.2.3	Accessibility of HELP	8-8
8.2.4	Provide HELP on HELP	8-8
8.2.5	Application Information	8-9
8.2.6	Provide HELP in Context	8-10
8.2.7	User Control of the HELP System	8-12
8.2.8	Provide Consistent HELP Format	8-12
8.2.9	Self-Explanatory and Concise Displays	8-12
8.2.10	Make Return to Application Easy	8-14
8.2.11	Keep HELP Current	8-14
8.2.12	Provide User Options	8-14
8.2.13	System-Initiated Messages	8-15
8.3	INTERACTIVE CONTROL	8-16
8.3.1	General	8-18
8.3.2	Context Definition	8-24
8.3.3	Transaction Selection	8-25
8.3.4	Interrupts	8-27
8.3.5	Error Management	8-30
8.3.6	Alarms	8-33
8.4	FUNCTION KEYS	8-33
8.4.1	General	8-34
8.4.2	Consistency	8-36
8.4.3	Double Keying	8-38
8.4.4	Labeling	8-38
8.4.5	Layout	8-38
8.4.6	Single Keying	8-40

TABLE OF CONTENTS (Cont.)

9.0	TEXT	9-1
9.1	TEXTUAL WINDOWS	9-1
	9.1.1 Data Field Labeling	9-1
	9.1.2 Updatable Fields	9-2
	9.1.3 Text Cursor	9-2
9.2	FORM FILLING	9-3
	9.2.1 General	9-3
	9.2.2 Defaults	9-4
	9.2.3 Consistency	9-5
	9.2.4 Cursor Movement	9-6
	9.2.5 Data Field	9-6
	9.2.6 Error Management	9-9
	9.2.7 Form Layout	9-10
	9.2.8 Labeling	9-14
10.0	GRAPHICS	10-1
10.1	MAPS AND SITUATION DISPLAYS	10-2
	10.1.1 General	10-2
	10.1.2 Static Display Attributes	10-4
	10.1.3 Dynamic Characteristics	10-8
	10.1.4 Display Editing	10-12
	10.1.5 Map Display Characteristics	10-13
10.2	PRESENTATION GRAPHICS (GRAPHS, PICTURES, AND DIAGRAMS)	10-14
	10.2.1 General	10-15
	10.2.2 Creating and Editing	10-16
	10.2.3 Scales, Labels, and Coding	10-19
	10.2.4 Identifying Critical Data	10-23
	10.2.5 Grid Lines	10-24
	10.2.6 Types of Presentation Graphics	10-25
	10.2.7 Pictures	10-29
	10.2.8 Diagrams (Schematics)	10-29
11.0	APPLICATIONS	11-1
11.1	DECISION AIDS	11-1
	11.1.1 When To Use Decision Aids	11-2
	11.1.2 Defining Decision Aid Requirements	11-9
	11.1.3 Features of Decision Aids	11-11
	11.1.4 User Requirements	11-13

TABLE OF CONTENTS (Cont.)

11.1.5	Organizational Factors	11-17
11.1.6	Flexibility	11-18
11.2	QUERY LANGUAGE	11-18
11.2.1	General Recommendations	11-21
11.2.2	Query Screen Design	11-23
11.2.3	User Requirements	11-27
11.2.4	User-Friendliness	11-29
11.2.5	Searching	11-31
11.2.6	Multiple Levels	11-35
11.3	EMBEDDED TRAINING	11-37
11.3.1	General	11-40
11.3.2	Adaptation to Users	11-43
11.3.3	Embedded Training Components	11-45
11.3.4	Instructional Structure	11-49
11.3.5	Instructional Presentation	11-49
11.3.6	Accessing Training	11-52
11.3.7	Screen Display	11-54
11.3.8	Technical Communication/Writing Style	11-54
11.3.9	Mobility/Navigation	11-54
11.3.10	Error Feedback	11-55
11.3.11	Ability to Modify	11-58

APPENDIX A - SECURITY PRESENTATION GUIDELINES	A-1
APPENDIX B - GLOSSARY	B-1
APPENDIX C - BIBLIOGRAPHY	C-1

LIST OF FIGURES

1.1	NIST User Interface System Reference Model	1-6
2.1	Motif and Open Look Scroll Bars	2-5
2.2	Comparison of Motif and Open Look Windows	2-8
3.1	Mouse Button Definitions	3-3
3.2	Large-Screen Display Technologies	3-17
4.1	Sample Workstation Screens	4-3
4.2	Tactical Information Display Options	4-7
4.3	Example of How Specific Types of Information Should Be Located on a Display Screen	4-9
4.4	Example of the Proper Location of Display Screen Instructions	4-10
4.5	Example of Breaking Information into Smaller Groups	4-10
4.6	Example of Vertical Data Presentation vs Horizontal	4-11
4.7	Example of How Data Should Be Justified	4-11
4.8	Example of Character Size and Spacing	4-13
4.9	Visual Arc Subtended	4-26
5.1	Example of a Typical Windowing Screen	5-1
5.2	Example of the Tiling Approach	5-2
5.3	Example of the Overlapping Approach	5-3
5.4	Example of Cluttered Window Design for Tiled Windows	5-12
5.5	Example of Cluttered Window Design for Overlapping Windows Induced by a Complex Background Pattern	5-12
5.6	Example of How a Dialog Box Should Be Designed	5-13
5.7	Example of Applications Menu With Menu Bars	5-14
5.8	Example of Window Animation	5-15
5.9	Example of a Screen Move	5-16
5.10	Example of a Cursor Changing Shape	5-17
5.11	Maximum Size of Window	5-18
5.12	Window Size Too Large, Covering Critical Information	5-18
5.13	Resize Border Removal	5-19
5.14	Example of Active Window Designation	5-20
5.15	Examples of Open Window Maps	5-22
6.1	Example of a Menu Bar	6-4
6.2	Example of a Pull-Down Menu	6-4
6.3	Example of a Pop-Up Menu	6-5
6.4	Example of Logical Ordering of Grouped Options	6-7

LIST OF FIGURES (Cont.)

6.5	Example of a System-Level Menu	6-9
6.6	Distinction Between Control Options and Command Options	6-10
6.7	Example of a Hierarchical Menu	6-11
6.8	Broad and Shallow Menu Tree vs. Narrow and Deep Menu Tree	6-12
6.9	Example of a Tree Diagram Interface	6-12
6.10	Example of a Combined Mode User Interface	6-13
6.11	Icons That Acknowledge Entry Selection	6-14
6.12	Example of Distinctive Subunit Labels	6-16
7.1	Example of Positive and Negative Images	7-4
7.2	Example of an Iconic Menu	7-7
7.3	Example of Textual Representation	7-9
7.4	Example of a Text Title for an Icon	7-10
7.5	Examples of Icon Shapes	7-10
7.6	Example of the Use of Technological Shapes	7-11
7.7	Example of Mirrored Icons	7-12
8.1	Printer Baud Rate	8-11
8.2	Example of How a Screen Display Can Be Customized	8-21
8.3	Example of How a System Can Display Default Status	8-26
8.4	Example of a General List of Control Options	8-28
8.5	Example of a Distinctive Confirm Action, Using a Dialog Box	8-32
8.6	Example of Soft Key Location	8-35
8.7	Suggested Method for Indicating Active and Inactive System Function Keys	8-37
8.8	Recommended Method for a Return to Base-Level Functions	8-37
8.9	Recommended Location for Function Key Labels	8-39
9.1	Example of Interrupt Capability for Multiple Entries	9-4
9.2	Example of How a Paper Entry Form and a Computer Data Entry Form Should Be Consistent	9-5
9.3	Cursor Should Appear in the First Character Space of the First Data Entry Field	9-7
9.4	Data Entry Should Not Require Leading Zeros	9-7
9.5	Visual Cues for Field Length	9-9
9.6	Example of How Multiple Screens Can Be Numbered	9-11
9.7	Example of How Explanatory Messages Can Be Provided	9-12
9.8	Example of a Form Title	9-13
9.9	Example of an Indication of an Optional Field	9-14

LIST OF FIGURES (Cont.)

10.1	Typical Electronic Map in Black and White	10-2
10.2	Example of Consistent Map Orientation	10-3
10.3	Brigade's Map Display Showing One Echelon Higher and Two Lower	10-5
10.4	Querying a Summary Symbol for Detailed Information	10-7
10.5	Example of a Map Inset	10-8
10.6	Example of How a Computer Model Can Generate Graphics from User Input	10-18
10.7	Texture Patterns	10-19
10.8	Example of Breaks in a Graph's Axes When Scales Have Been Expanded	10-20
10.9	Example of the Distortion Induced in Data Trends by Exaggerated Scales	10-21
10.10	Example of Linear Versus Non-Linear Scales	10-21
10.11	Example of Supplementary Text and Actual Value Use in a Graph	10-23
10.12	Example of a Surface Chart	10-26
10.13	Example of Bar Graphs	10-27
10.14	Example of a Scatterplot	10-27
10.15	Example of a Pie Chart	10-28
10.16	Example of a Graphic Picture	10-30
10.17	Sample Flowchart	10-31
11.1	Examples of Pie Chart, Bar Graph, and Line Graph	11-23
11.2	Data Layout and Justification	11-26
11.3	Sample Text Editing Box	11-32
11.4	Simplified Prototype Embedded Training Screen	11-40
11.5	Examples of Implying Human Characteristics	11-41
11.6	User Selection of Training Level	11-44
11.7	Example of How an Advisor Can Guide Users	11-46
11.8	Example of "Cautions" that Identify Common Errors	11-48
11.9	Combined Graphic and Natural Language Presentation	11-50
11.10	Example of Assistance Request Reminder	11-51
11.11	Using an Icon to Access Embedded Training Directly	11-53
11.12	Examples of Tone in Error Message	11-57
11.13	Example of Automatic Correction Notification and Identification	11-58
11.14	Example of Identification Annotation Position	11-59
A.1	Sample CMW Screens	A-8
A.2	Sample Trusted Path Main Menu	A-11
A.3	Sample for Entering New Input Information Label	A-13

LIST OF TABLES

2.1	Examples of Terminology Differences	2-2
3.1	Required Pixels for Stroke Width	3-7
3.2	Required Pixels for Width Design	3-8
3.3	Maximum Luminance Nonuniformity	3-8
3.4	Image Formation Time Classes	3-9
4.1	Definitions of Key Terms for Color Usage	4-16
4.2	Recommended Luminance Contrast Ratios	4-20
4.3	Color Codes and Associated Meanings	4-23
4.4	Classification Bar Color Codes and Associated Meanings	4-24
6.1	Examples of Option Codes	6-17

1.0 INTRODUCTION

1.1 BACKGROUND

The proliferation of computer technology has resulted in the development of numerous computer-based systems and the implementation of varying styles of the Human Computer Interface (HCI). To accommodate the expected growth in computer-based systems, minimize HCI diversity, and improve system performance and reliability, the U.S. Department of Defense (DoD) is adopting software development standards and providing guidelines for information display and manipulation.

Computer-based system performance and reliability are products of the performance and reliability of each individual component. Computer-based system components include hardware, software, and any user involved in operation, maintenance, or utilization of the system. Of these components, the user is the most important as well as the most difficult to predict. Thus, a key factor of a high performance, high reliability system is an easy-to-use, effective design of the interface between the user, the hardware, and the software.

HCI standardization is based on selection of a Graphical User Interface (GUI), which in turn provides a standard Application Programming Interface (API) and style approach. The GUI is determined by the software source selected such as Commercial-Off-The-Shelf Software (COTS), Government-Off-The-shelf Software (GOTS), or proprietary software applications. The variability of users' needs and differing interpretations of GUI style result in the lack of a common approach and the creation of dissimilar HCIs among applications developed by independent organizations. Adding to standardization problems is the fact that most of these packages do not address several issues common to DoD systems, for example, map graphics operations.

Standardizing the HCI across application software developed within the DoD community is a two-step process. The first step is the definition and documentation of the functional goals, objectives, and requirements of the HCI. The second is the understanding of HCI standardization benefits, and the acceptance and adoption of those requirements by DoD application designers.

1.2 PURPOSE

The purpose of this *Style Guide* is to provide a common framework for HCI design and implementation. Through this framework, the long-term functional goals, objectives, and requirements of the HCI will be defined and documented. Interface implementation options will be standardized, enabling all DoD applications to appear and operate in a reasonably consistent manner.

Specifying appearance, operation, and behavior of DoD software applications will support the following operational objectives:

- Higher Productivity - People will accept and use what is easy to understand if it aids them in accomplishing their assigned tasks with minimal confusion or frustration.
- Less Training Time - Standard training can be given once for all applications, rather than once for each application.
- Reduced Development Time - It will no longer be necessary to design a complete HCI for each system component, as previously developed *Style Guide*-compliant software will be available. The basic appearance and behavior of the interface will be specified by this *Style Guide*. The specific "look and feel" of each DoD organization's applications software will be detailed in attachments. These details will limit HCI diversity, and further support the reduction of development time for unique HCIs.

1.3 BASELINE

This *Style Guide* has been prepared after a review of current HCI research and standardization efforts, including the following:

- The National Institute of Standards and Technology (NIST) User Interface System Reference Model, as found in The Technical Reference Model for Corporate Information Management (DISA/CIM 1991 Draft), has been adopted as the baseline for this document.
- The Department of Defense Intelligence Information Systems (DODIIS) Style Guide, (DODIIS 1991) from which Version 1.0 of this *Style Guide* was adapted.

- Federal Information Processing Standard (FIPS) 158, "User Interface Component of Applications Portability Profile," (NIST 1990b) which mandates the use of the X Window protocol, X library, and X toolkit intrinsics.
- Air Force Intelligence Data Handling System Style Guide (U.S. Air Force 1990) establishes HCI guidelines for applications developed for Air Force Intelligence analysts and users.
- The Defense Intelligence Agency (DIA) Standard User Interface Style Guide for Compartmented Mode Workstations (DIA 1983, henceforth called the DIA Style Guide) and Compartmented Mode Workstation Labeling: Source Code and User Interface Guidelines (DIA 1989, henceforth called DDS-2600-6215-89). These documents address the security portion of the HCI and are intended for designers of applications for Compartmented Mode Workstations (CMWs). They outline security-related interface requirements for workstations operating in the system high or compartmented mode.
- The Open Look Graphical User Interface Application Style Guidelines (Sun Microsystems, Inc. 1990, henceforth called the Open Look Style Guide) and the Open Software Foundation (OSF)/Motif Style Guide (OSF 1990), which describe the two major X Window GUIs.
- North Atlantic Treaty Organization Standardization Agreement 2019, Military Symbols for Land Based Systems (NATO 1990); Army Field Manual 101-5-1, Operational Terms and Symbols (U.S. Army 1985b); and "DIA Standard Military Graphics Symbols Manual" (DIA 1990 Draft), which standardize map graphics symbols.
- Human Factors Guidelines for the Army Tactical Command and Control System Soldier-Machine Interface, Versions 1.0 and 2.0 (Avery et al. 1990 and 1992), which provide a set of overarching guidelines focused on designing the user-computer interface to enhance user performance.
- "Institute of Electrical and Electronics Engineers (IEEE) Recommended Practices for Graphical User Interface Drivability," Draft (IEEE 1992b, henceforth called IEEE P1201.2) . When adopted, this document will standardize those HCI elements and characteristics that must be consistent to facilitate users switching from one "look and feel" or application to another.
- The User Interface Specifications For Navy Command And Control Systems, Version 1.1 (Fernandes 1992), which defines a common "look and feel" for Navy command and control systems.

- Human Engineering Design Criteria for Military Systems, Equipment and Facilities, MIL-STD-1472D (DoD 1989b) and Human Engineering Guidelines for Management Information Systems, DOD-HDBK-71A (DoD 1989c), both of which are human factors standards currently in use by DoD.

This *Style Guide* extends and, in some instances, includes by reference the aforementioned documents. The intention is to establish style objectives and guidelines to which all members of the DoD community can transition.

1.4 HUMAN-COMPUTER INTERFACE

The user is an integral part of a system, and the user-machine interface encompasses any interaction between the user and the system. This includes controls, displays, environmental concerns (e.g., lighting and noise), workspace layout, procedures, and documentation. The design of these elements has a major impact on manpower, personnel selection, training, logistics, safety, and human performance, all which are elements of concern within DoD systems. HCI addresses the user interface as applied to computer-based systems. HCI encompasses the "look and feel" of the interface, physical interaction devices, graphical interaction objects, alternate interactions (i.e., voice, touch screen, pen, etc.), environmental factors, and any other human-computer interactive methodology.

Design guidelines for the HCI provide three major benefits:

- First, designers have a resource upon which to draw to aid in developing usable display screens and interactive procedures. This is especially important with the rapid pace of knowledge acquisition on human performance and computer systems, and because graphical user interface (GUI) is emerging as the dominant architecture for the human-computer interface (HCI).
- Second, the guidelines provide a common approach that supports consistency of design, which is a fundamental principle of human factors engineering design.
- Third, the guidelines will allow less stringent personnel selection criteria, and will reduce training and possibly manpower requirements for all systems.

1.5 SCOPE

Two factors influence the applicability of the *Style Guide* to military computer-based systems: the software architecture being used and the functional requirements of the target system. This document addresses functional requirements and operations that should be consistent across the entire HCI, with emphasis on HCI considerations for features and functions applicable to DoD applications. Such features and functions include system start-up, security issues, and map graphics.

The *Style Guide* has been developed to address design considerations germane to the DoD environment. The guidelines are generic enough to be applicable to almost any GUI and, to a lesser extent, text-based interfaces. The system developer needs to be aware that using a software architecture other than those mandated for use within DoD will limit portability to and reusability by other systems within DoD.

Guidelines are presented for application development within layers 0 through 5 of the NIST reference model. A summary of the NIST reference model is presented in Figure 1.1. For layers 0 through 2, applications should adhere to the X Window processing standards in FIPS 158. Layer 3 defines the toolkit standards that support the window management API. Layers 4 and 5 define the "look and feel" of the GUI. Standards for the upper layers are currently being developed by the IEEE P1201 committees and may eventually be incorporated into this *Style Guide*.

1.6 INTENDED AUDIENCE

The target audience for this document includes military and civilian personnel representing program management, system managers, and application HCI designers. Ideally, these individuals should be knowledgeable of the characteristics of the intended user population and the tasks they must perform. In addition, the users of this *Style Guide* should have some knowledge of human-performance considerations. A secondary audience includes users and software maintainers who are interested in the general design of the interface or who wish to provide feedback on the *Style Guide*.

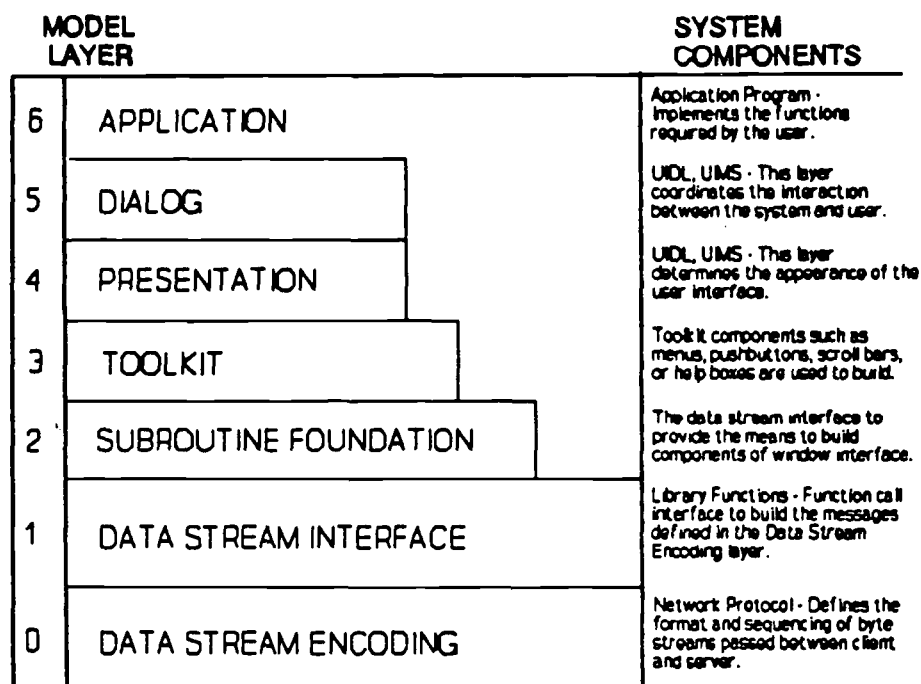


FIGURE 1.1: NIST User Interface System Reference Model

1.7 DESIGN GOALS

DoD application development should achieve the following objectives:

- First, the functions and tasks to be performed by the system and the operational environment must be known. This allows development of an understanding of the overall system dynamics.
- Second, an analysis of the capabilities and limitations of the system users must be completed. This allows development of an understanding of which tasks are best performed by the human and which are best performed by the hardware and software. In addition, this understanding provides the groundwork for task and interface design to ensure the user can successfully perform the required tasks.
- Finally, a consistent set of rules for designing the interface must be applied. The design of the HCI includes, but is not limited to, the following:

- a. Applications should be designed to meet the specific requirements of the user. Above all, provide the functionality to meet those requirements.
- b. All applications should be consistent with the interface guidelines specified in this document.
- c. An application's HCI should provide rapid access to all of its functions. To ensure this, avoid unnecessary menus and long selection lists that force users to "page" through all entries.
- d. An application should be flexible. For example, provide multiple methods to access a function (e.g., direct command line entry, menus, tree diagrams, mnemonics, and keyboard accelerators).
- e. Explicit action should be required to perform any act that could result in irreversible negative consequences and users should be provided options (e.g., quit without saving).
- f. Users should have a choice of input devices (keyboard or pointing device) for scrolling, map manipulation, and invoking or terminating an application. The keyboard and pointing device should be interchangeable where appropriate to the action being performed.
- g. An application interface should not depend on color to communicate with the user. Color should add substance to the interface, not dominate it.

1.8 **ASSUMPTIONS**

In writing this *Style Guide*, the following assumptions were made:

- The user will be interfacing with message-handling systems, COTS software, and GOTS applications.
- The application design requirements specified in this *Style Guide* will be supported by standard DoD civilian computer environments, and tactical or strategic computer environments. The DoD HCI will be implemented on a variety of computer architectures. Computer systems will be equipped with diverse capabilities, such as monochrome versus color monitors and varying amounts of random access memory.

- The *Style Guide* will not address all elements of the man-machine interface. The focus of this document is on the HCI within DoD.

1.9 STYLE GUIDE ORGANIZATION

Section 2 of the *Style Guide* describes the primary differences in the terminology used to describe Motif and Open Look features and functions, and the major behavioral differences in applications written for the Motif and Open Look interfaces.

Section 3 describes hardware considerations, with focus on input/output devices and their alternatives. This section will address issues related to the Computer/Electronic Accommodation Program (CAP). A subsection on special displays is also included.

Sections 4 through 10 contain HCI guidelines for the designer. The general subjects covered include screen design, windows, menu design, object orientation, common features, text, and graphics. Each of these sections is divided into specific subject areas and include examples of the stated design guidelines.

Section 11 covers application design guidelines. The topics include decision aids, query, and embedded training. The selected applications represent focus areas of DoD applications and subjects that have generated questions and comments from system developers.

Appendix A describes objective security interface requirements, using the DIA Style Guide and DDS-2600-6215-89 as baselines.

Appendix B, the glossary, defines frequently used terms pertaining to the HCI and GUI style guidelines.

Appendix C, the bibliography, is supplemented by direct references at the end of each section, which provide a means of determining the original source of a specific guideline. Appendix C demonstrates the overall review undertaken to provide a baseline for this document.

Addenda will describe specific interface requirements of various organizations served by this style guide. This version of the *Style Guide* will include "User Interface Specifications for Navy Command and Control Systems" as Addendum 1. Additional addenda will be added as required.

2.0 MOTIF AND OPEN LOOK STYLISTIC DIFFERENCES

The DoD software architecture includes systems using both Open Look and Motif GUI styles. The *Style Guide* has been developed to address design considerations germane to both style environments. This section describes the primary differences in the terminology used to describe Motif and Open Look features and functions, and the major behavioral differences in applications written for the Motif and Open Look interfaces. Regardless of the interface, applications should adhere to the X Window processing standards in FIPS 158 (NIST 1990b).

The OSF/Motif and Open Look GUIs contain numerous stylistic differences. This sections groups these differences into the following categories:

- Terminology - differences in the names assigned to, and descriptions of, basic Motif and Open Look functions and features.
- Look - differences in the appearance of Motif and Open Look displays.
- Feel - differences in the actions a user takes to interact with Motif and Open Look applications.

2.1 DIFFERENCES IN TERMINOLOGY

Motif and Open Look style guides use substantially different terms to describe the functions and features associated with the respective GUIs. The main distinction is that different terms and descriptive phrases are used to define and describe equivalent or similar functions and features. However, there are also instances in which the same term is used to refer to different unrelated functions or features. Table 2.1 identifies some of the primary terminology differences, and maps the terms and descriptive phrases used in Motif to those used in Open Look. Although the mapping is sometimes to equivalent capabilities, the capability supported by one GUI is often only a subset or approximation of that supported by the other.

An example of using different terminology to describe similar functions is the Motif use of the term "radio button" and the Open Look term "exclusive setting." Both terms refer to lists of selections for which only one choice can be made.

TABLE 2.1: Examples of Terminology Differences

<u>Motif Term</u>	<u>Relationship</u>	<u>Open Look Term</u>
Menu Bar	Is Similar To	Control Area
Cascading Menu	Is Equivalent To	Submenu
Dialog Box	Is Similar To	Pop-Up Menu
Message Box	Is Equivalent To	Notice
Scroll Bar (on Slider)	Is Similar To	Proportion Indicator
Stepper Arrow (on Slider)	Is Equivalent To	Scrolling Arrow
Scroll Region (on Slider)	Is Similar To	Drag Box
Dialog Box Routing	Is Equivalent To	Window Item
Cascading Menu Routing	Is Equivalent To	Menu Item
Accelerator	Is Similar To	
Accelerator/Mnemonic		
Mnemonic	Has No Counterpart	-----
Radio Button	Is Equivalent To	Exclusive Setting
Check Button	Is Equivalent To	NonExclusive
Setting		
Push Button	Is Equivalent To	Command Button
-----	Has No Counterpart	Pushpin
Action	Is Equivalent To	Command
Command	Has No Counterpart	-----
Close	Is Equivalent To	Quit
Minimize	Is Equivalent To	Close
Maximize	Is Equivalent To	Full Size

An example of using the same terminology to mean something completely different is the term "command." A command window in OSF/Motif refers to a window which allows users to enter commands to the application or the operating system; in Open Look, a command window is used to set parameters and execute commands based upon those parameters.

Terminology differences are especially significant in the display of menu items

and window commands. Because terms used in Motif and Open Look applications will correspond to the terminology in the respective style guides, the line item entries for similar actions will differ.

Terminology differences will also affect the documentation for common applications. Documentation developed for Motif-based applications will use terms specific to Motif environments, and Open Look documentation will use terms specific to Open Look environments.

2.2 DIFFERENCES IN APPEARANCE

Some of the primary differences related to the appearance of Motif and Open Look displays are as follows:

- Motif displays use shading to simulate a 3-dimensional effect; early versions of Open Look are based upon 2-dimensional representations.
- Motif uses wide resize borders (consisting of edge and corner handles) to signify that the size of the window can be changed; Open Look uses resize corners to denote this capability.
- Motif makes extensive use of settings (check buttons and radio buttons) and Open Look emphasizes command buttons.
- Motif uses an arrow to denote menu items in menu lists; Open Look uses a hollow triangle.
- Motif Title Bars contain minimize and maximize buttons; Open Look Title Bars do not.
- Open Look displays a Pushpin on Pop-Up Windows to give users the option of retaining the window for repeated operations; Motif does not support this feature.
- Motif menus normally display mnemonics and accelerators; Open Look menus do not normally display this information.
- Motif icons are placed in a box or area, normally at the lower left-hand corner of the screen; Open Look does not use an icon box.
- In Motif applications, HELP is available as a menu option (shown on the menu

bar) and through a keyboard accelerator; in Open Look, HELP is accessed through a help key.

- Motif and Open Look scroll bars have different representations (see Figure 2.1).

2.3 DIFFERENCES IN FEEL

Differences in the feel of Motif and Open Look interfaces relate to the definition and use of the mouse buttons; the definition of keyboard special-purpose keys, mnemonics, and accelerators; and the use of some special-purpose controls.

2.3.1 MOUSE BUTTON DEFINITION DIFFERENCES

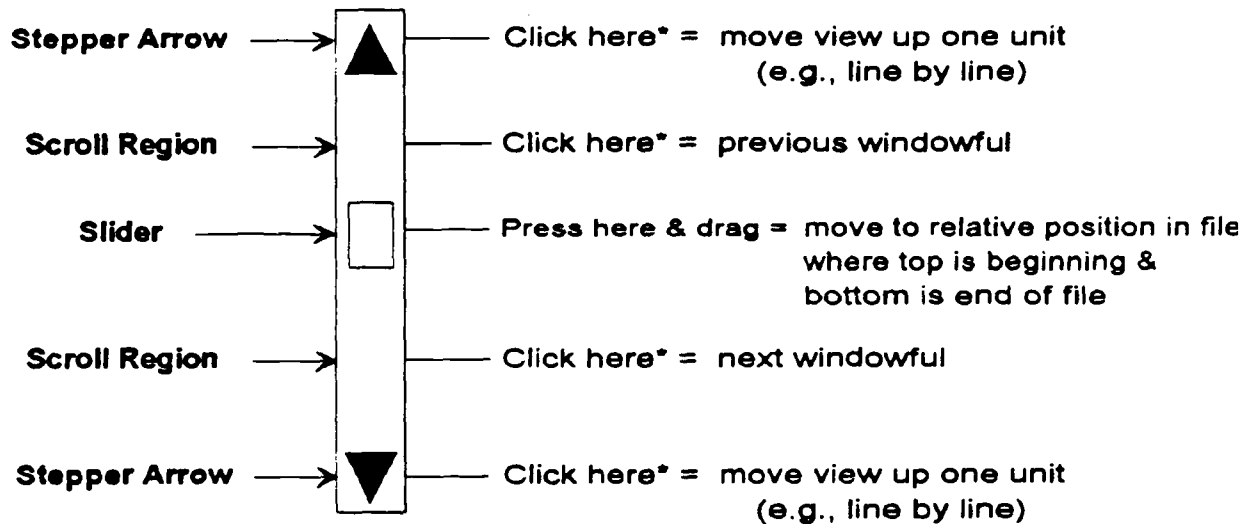
The default button definitions for two- and three-button mice are different in Motif and Open Look implementations. For two-button mice, Motif accommodates three functions: SELECT, DRAG, and CUSTOM. Open Look only accommodates SELECT and MENU functions on the mouse, while the the ADJUST function is implemented using the keyboard shift key in conjunction with the mouse-SELECT button.

For three-button mice, Motif defines the middle button as the DRAG button and the outside button as the CUSTOM button; Open Look defines the middle button as the ADJUST button and the outside button as the MENU button. Both Motif and Open Look define the inside button as the SELECT button. The most significant problem related to the difference in mouse button definitions is not the fact that the button definitions seem to be reversed (i.e., DRAG/CUSTOM versus ADJUST/MENU), but rather the fact that the Motif and Open Look implementations do not use the buttons in a consistent manner.

The SELECT button in Motif implementations is used to choose objects and display menus. However, in Open Look applications, the SELECT button is used to choose objects and to manipulate objects and controls; the MENU button is used to display menus. In Motif applications, the DRAG button is used for the direct manipulation of objects. In other words, some functions performed with the SELECT button in Motif applications will be performed with the MENU button in Open Look applications and conversely, some functions performed with the DRAG button in Motif applications will be performed with the SELECT button in Open Look applications. To make matters a little more confusing, the Motif CUSTOM button can be used to display pop-up menus, whereas this function is performed in Open Look applications using the MENU button. Additionally, the function supported by the Open Look ADJUST button (i.e., to extend or reduce the number of selected objects) is implemented in

Motif applications using the SELECT button. In short, the use of the mouse buttons will fluctuate depending upon whether one is interacting with a Motif or

Motif



Open Look

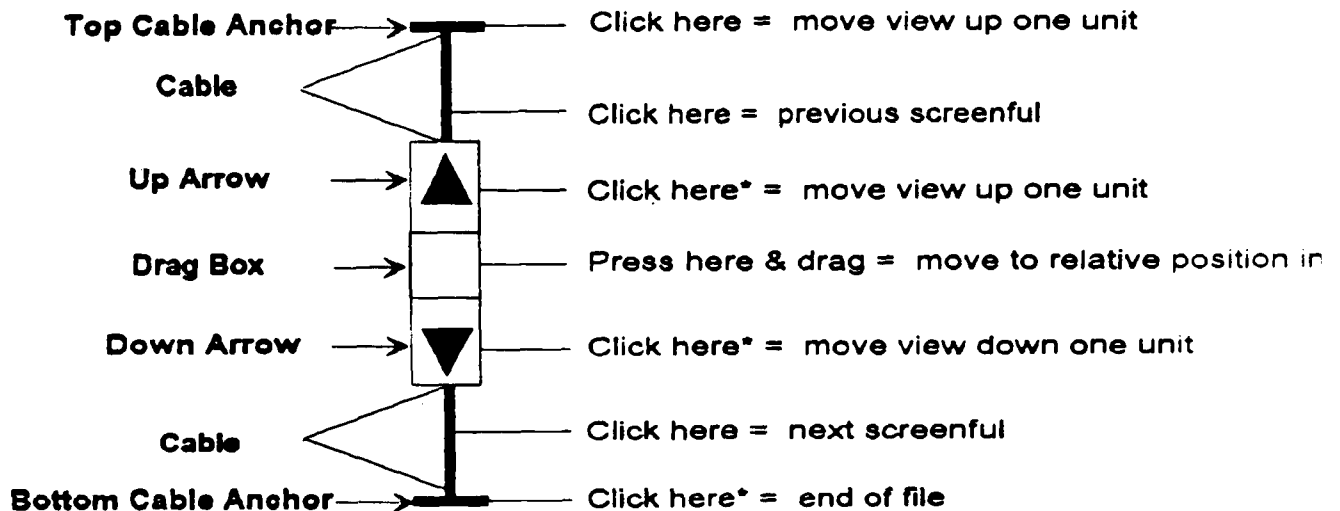


FIGURE 2.1: Motif and Open Look Scroll Bars

Open Look application.

2.3.2 KEYBOARD USAGE DIFFERENCES

Both Motif and Open Look support keyboard input, but there is very little consistency between the two GUIs in the definition of special-purpose keys. For example, the Motif Style Guide contains an extensive list of default keyboard function assignments with definitions for function keys, navigation keys, and special-purpose keys (e.g., Esc). The list includes definitions for keys used with and without modifiers (i.e., Shift, Control, and Alt keys). Accelerators and mnemonics are also defined for Window Management Functions, and default key assignments are made for entering data in text entry boxes. The list of default key assignments in the Motif Style Guide is quite extensive.

The Open Look Style Guide also includes guidelines for using the keyboard, but the default key assignments are not nearly as extensive as in the Motif Style Guide. In essence, the Open Look Style Guide does not emphasize keyboard use as much as the Motif Style Guide, and rather than suggesting default assignments for special-purpose keys, such as function keys, the Open Look guidelines actually state that developers should minimize the use of these keys. For example, the Open Look Style Guide states that the use of function keys should be avoided for two primary reasons: 1) there are numerous differences in the number of function keys supported by various manufacturers' keyboards; and 2) some keyboards do not have function keys. Open Look does not rule out the use of special-purpose keys, however, allowing software developers to make the final decision regarding their definition.

The main problem related to keyboard usage within Motif and Open Look applications is that it is unlikely the definitions for special-purpose keys and accelerators will be consistent among applications developed for each GUI. Because software developers will use the key assignments and guidelines specified in the respective GUI style guide, users will be faced with numerous keyboard implementation differences when interacting with Motif and Open Look applications. Because of such differences, users who make extensive use of the keyboard may find it difficult to adjust between Motif and Open Look applications unless they develop templates or redefine the keyboard mappings.

Another inconsistency between Motif and Open Look interfaces is in the implementation of mnemonics. In Motif, menu mnemonics are implemented by typing a single letter (i.e., the underlined letter within a menu line item); in Open Look, the typing of a mnemonic requires the use of a modifier, such as the control key, in conjunction with the mnemonic letter. In particular, Open Look

mnemonics are similar to Motif accelerators rather than Motif mnemonics. Open Look mnemonics are also restricted to the first letter of a menu entry; Motif mnemonics, which are underlined, can be any letter within the menu item.

2.3.3 DIFFERENCES IN THE USE OF CONTROLS

The most significant differences between controls used in Motif and Open Look implementations are found in the use of the scroll bar and menu bar (Control Area).

2.3.3.1 Scroll Bar Differences

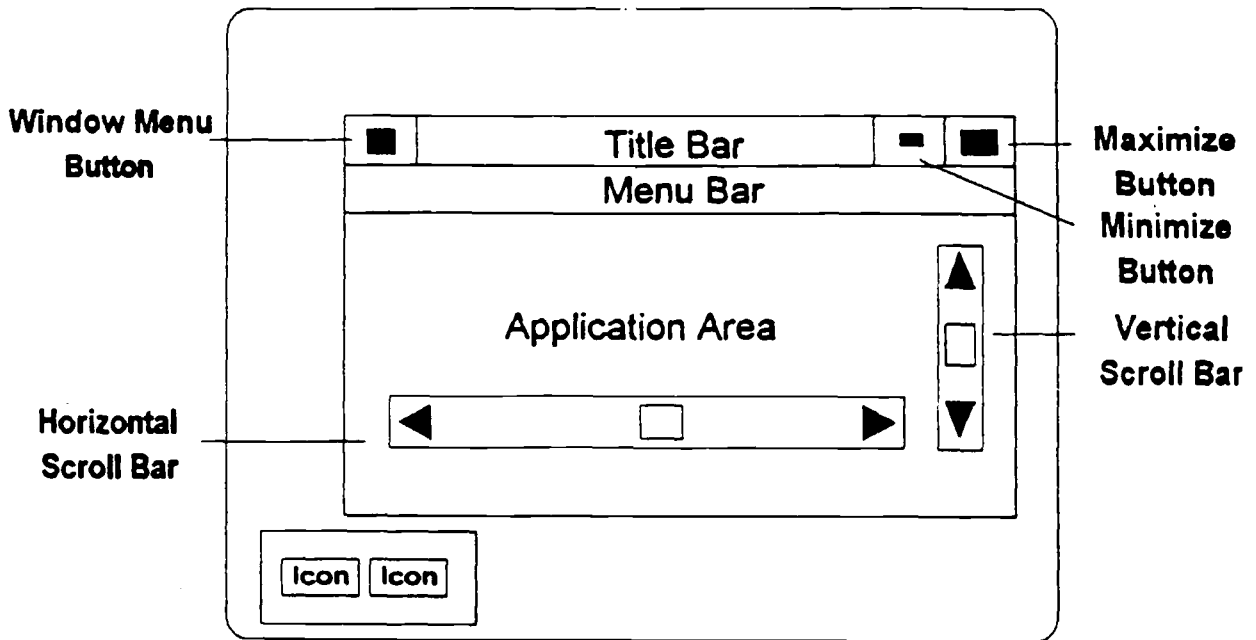
In addition to differences in the terminology used to define Motif and Open Look scroll bars (see Table 2.1), there are also differences in the look of the scroll bars (see Figure 2.1). These stylistic variations also effect differences in the features supported by the two scroll bars. For example, in Open Look, there are two ways to position the pointer at the top and bottom of a file; in Motif there is only one. In Open Look, clicking on the Top Cable Anchor in the scroll bar will move the pointer to the beginning of the file, clicking on the Bottom Cable Anchor will move the pointer to the end of the file. The pointer may also be positioned at the beginning or end of a file by moving the Proportion Indicator to the top or bottom of the scroll bar. Moving to the beginning or end of a Motif file is done by positioning the Slider at the top or bottom of the scroll bar.

Another difference in scroll bar functionality is that Motif implementations allow users to continuously page through a file by pressing the SELECT button in the scroll region (i.e., above or below the Slider). In Open Look applications, fast paging requires multiple click operations. Additionally, Open Look applications have the option of generating page-oriented scroll bars that display the current page number of a file next to the Drag Box; Motif applications do not.

2.3.3.2 Window Menu Bar/Control Area Differences

As noted in Figure 2.2, the line beneath the Title Bar of the base window is referred to as the Menu Bar in Motif implementations and the Control Area in Open Look. In addition to the differences in terminology, there are also differences in functionality. In Motif applications, the Menu Bar is usually made up of lists of menu topics (push buttons are also allowed), resulting in

Motif



Open Look

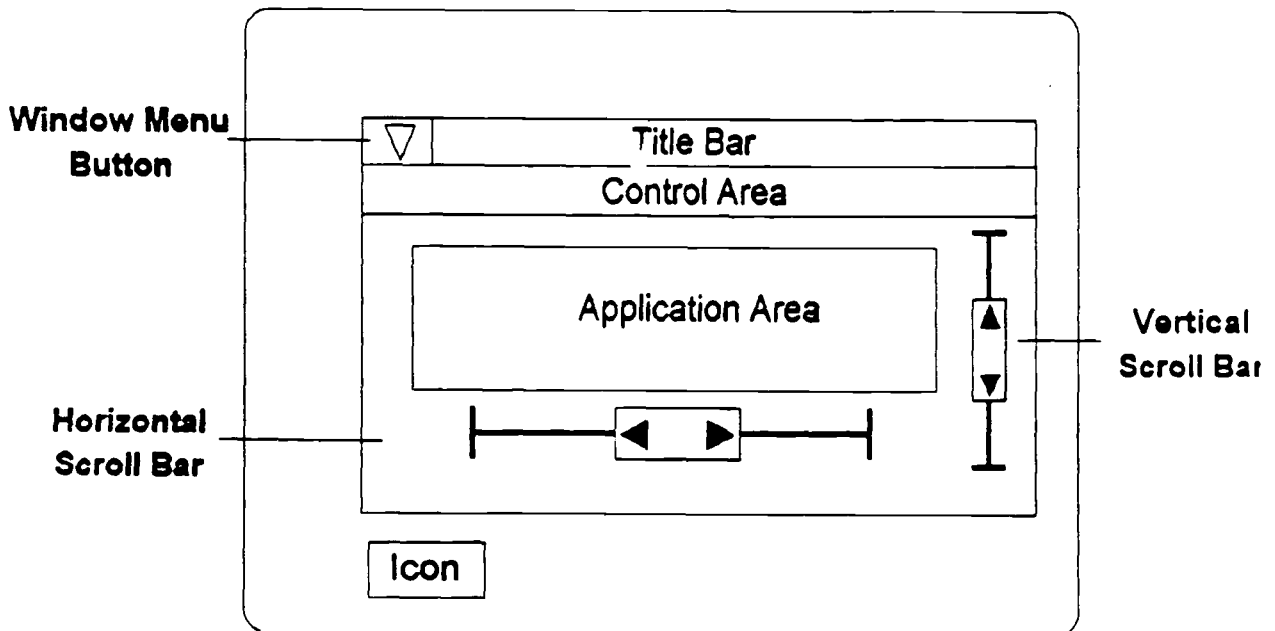


FIGURE 2.2: Comparison of Motif and Open Look Windows

the display of pull-down menus when they are selected. In Open Look applications, the Control Area can contain controls of any type, including such items as buttons, settings, check boxes, text fields, sliders, and gauges. Open Look allows developers to include frequently used commands in the Control Area.

2.4 STYLE GUIDE PRESENTATION

The illustrations used in the *Style Guide* are based upon the Motif GUI. The comparisons in this section will allow the designer to note the differences in the Open Look GUI and apply the stated guidelines. The examples used in this document are based upon actual applications and are intended as guides rather than templates. The primary sources for examples have been operational applications from the intelligence or the command and control communities. The design guidance can be applied to all layers of application and all types of GUI styles.

2.5 CONCLUSION

There are many stylistic differences between the Motif and Open Look GUIs. The significance of these differences will vary among users but will become most difficult to deal with when systems are not standardized on a single GUI. The designer shall not use more than one GUI style within a single application/system.

3.0 HARDWARE

Hardware refers to the computer and all supporting devices that impact the HCI. It is difficult to develop standard guidelines for all possible hardware variations within DoD, primarily because of the differences in users' requirements. Hardware requirements can vary extensively, depending on the function being performed by the system. Some systems are actually information management systems and business systems that do not require immediate user response to information available through the interface. On the other hand, real-time tactical display and control systems require the user to make immediate decisions and to input commands from the information on the interface. Each system has different hardware and interface design requirements based on its primary function. The designer needs to understand the selected hardware and the primary function of the system being developed to provide an effective HCI.

Section 3.1 will highlight the procedures used to communicate with system applications using a pointing device or the keyboard.

The purpose of Section 3.2 is to present guidelines relevant to the specification, selection, use, or design of displays other than the Cathode Ray Tube (CRT). Section 3.2 has been divided into subsections, each of which describes current technology, cites advantages and limitations, and presents available guidelines. The subsections deal with five types of special display technology:

- flat-panel displays
- large-screen displays
- stereographic and 3D displays
- glare reduction techniques
- Touch Interface Devices (TIDs).

Section 3.3 is entitled "Alternate Input/Output (I/O) Devices." This section addresses the area of non-standard access to a GUI environment. The guidelines in this section will consider the requirements of the Computer/Electronic Accommodation Program (CAP).

3.1 INPUT DEVICES AND PROCEDURES

Section 3.1 will highlight the procedures used to communicate with system applications using a pointing device or the keyboard. A comparison is made between Motif and Open Look to illustrate the impact of GUI style selection on hardware. For a more detailed explanation of the input procedures, consult the appropriate GUI style guide.

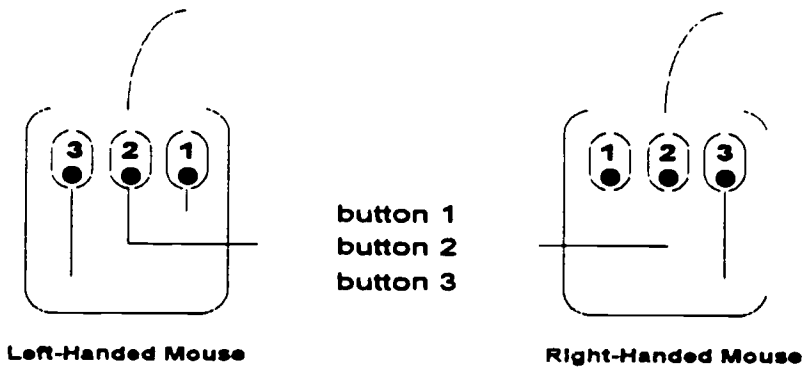
3.1.1 POINTING DEVICES

A pointing device (e.g., mouse, trackball, tablet, or light pen) allows a user to navigate rapidly around the screen and to specify and select objects for manipulation and action. Throughout this *Style Guide*, the mouse is used as the prototype for all pointing devices.

3.1.1.1 Mouse Button Definitions

Within the DoD community, both two-button and three-button mice are used. Figure 3.1 illustrates the Motif and Open Look default button definitions for both types. For users who must interact with both Motif and Open Look applications, it is important to notice that the button definitions and button use differ. Differences are detailed in Section 2. The mouse button operations supported by both GUIs are consistent, and can be defined as follows:

- **Press** - pushing the mouse button and holding it
- **Release** - letting up on the mouse button
- **Click** - quickly pushing and releasing a mouse button before moving the pointer
- **Double-click** - pushing and releasing the mouse button twice in quick succession
- **Move** - sliding the pointer without pushing any mouse buttons
- **Drag** - pushing the mouse button and holding it while moving the pointer



MOTIF

3-Button Mouse	2-Button Mouse	Name	Function
button 1	button 1	Select	Select objects; display pull-down menus
button 2	button 1	Drag	Manipulate objects (e.g., moving, dragging)
button 3	both buttons simultaneously	Custom	Display pop-up menus; application-specific functions

OPEN LOOK

3-Button Mouse	2-Button Mouse	Name	Function
button 1	button 1	Select	Select objects; manipulate objects & controls
button 2		Adjust	Extend or reduce number of selected objects
button 3	button 2	Menu	Display menus

FIGURE 3.1: Mouse Button Definitions

To "drag an object with the mouse" is to move the pointer over the object, press the SELECT button on the mouse, move the mouse until the object is in the desired location, then release the SELECT button.

3.1.1.2 The Pointer

A key element of the workspace is the pointer. Objects on the screen can be manipulated by positioning the pointer over the object and pressing the mouse buttons appropriately. The user moves the pointer by moving the

mouse on the mouse-pad. (Note: Different actions are used to move other pointing devices, such as trackballs and light pens.)

Mouse pointer shapes provide visual clues to the activity within a window. For example, an hourglass or watch-shaped pointer can be used to indicate that an application is busy, and a cross-hair can be used when sighting on a graphics display.

With the exception of applications using computer-controlled tracking, the pointer should remain where it is placed until moved by the user or the application. The pointer should not "drift."

3.1.2 THE KEYBOARD

The keyboard should be virtually interchangeable with the mouse to allow a user to interact with the application by using a pointing device, the keyboard, or both. Although keyboards vary greatly in the number and arrangement of keys, most keyboards include the following:

- **Alphanumeric Keys** - Letters of the alphabet, numbers, punctuation symbols, and text-formatting functions (e.g., Tab, Return, Space Bar)
- **Modifier Keys** - Keys (typically Shift, Control, and Alt) that modify or qualify the effect of other keys (or pointing device inputs) for as long as they are held down
- **Navigation Keys** - Keys that are used to move the cursor (e.g., arrow keys, Home, End, Page Up)
- **Function Keys** - Keys (typically F1 through F10) provided for extra or general functions
- **Special-Purpose Keys** - Keys that have a special function (e.g., Help, Delete, Escape, Backspace, Insert, and Enter)

A function should not be solely available through a function key because, keyboards differ and function keys vary according to application and GUI. Guidelines for OSF/Motif and Open Look application key assignments are provided in the respective GUI style guides.

3.1.3 WINDOW INPUT FOCUS

Usually, several application windows are ready to accept input, but only one window, the one with "input focus," actually receives the user input. The window with input focus is known as the active window and is the window where keyboard input appears and input from the pointing device apply.

Most interfaces provide explicit input focus; that is, the user (or application) performs an action (e.g., typing appropriate keyboard accelerators, clicking a pointer inside a window, or moving a window to foreground through menu selection) to assign input focus. Implicit focus (the focus is automatically assigned to the window containing the location cursor) is often provided as an option. The default for both Motif and Open Look applications should be explicit focus.

A window with input focus should move the window to the front of the workspace and highlight the window in some fashion, such as highlighting the window frame or title bar.

3.2 SPECIAL DISPLAYS

The CRT is the principal display technology used in computer-based systems. Success of the CRT can be attributed to its ability to deliver inexpensively full-color imagery at high luminance and resolution. However, tasks of the modern military and emerging alternative display technologies have permitted development of computer-based systems using display technology other than the traditional CRT. Examples include the Liquid Crystal Display (LCD) used in portable computers and projection technology used to brief military personnel in command centers. As a result, the designer has more alternatives when selecting a display for a military system. The purpose of Section 3.2 is to present guidelines relevant to the specification, selection, use, or design of displays other than the CRT.

Section 3.2 is divided into subsections, each of which describes current technology, cites advantages and limitations, and presents available guidelines. Many reports upon which this section is based were published in various conference proceedings rather than in refereed scientific journals and, therefore, may have had less extensive peer review and professional scrutiny. The *Style Guide* user should consult the references for further information on display technologies and human performance considerations.

3.2.1 FLAT-PANEL TECHNOLOGY

A flat-panel display is flat and light and does not require a lot of power. "Flat" means being thin in form, as well as having a flat display surface. An ideal flat-panel display has the following characteristics: thin form, low volume (cubic size), even surface, high resolution, high contrast, sunlight readable, color, low power, and light weight. Recent advances in flat-panel display technologies have made them realistic alternatives to CRTs for displaying information at computer workstations. Advances have been made in many areas: addressability, contrast, luminance, and color production. Continued research in flat-panel displays has resulted in introducing high information content products that challenge the CRT in specialized applications.

Although a long list of flat-panel display types can be made, LCDs, electro-luminescent displays, and gas plasma displays are the only flat-panel technologies currently mature enough and economical enough to be used in the military. A major characteristic of each of these display technologies, as distinguished from CRT technology, is that images are formed by turning discrete, non-overlapping, rectangular, cell-based pixels on and off. This discrete, pixel-based structure provides part of the reason that measures of image quality used to evaluate CRT resolution cannot be effectively used to predict image quality and human performance with flat-panel displays.

Factors affecting human performance that differ from the guidance given for CRTs include character-to-character spacing, interline spacing, character and symbol design, the effect of ambient illumination, image polarity, and failure mode. An overriding guideline when specifying flat-panel display technology relative to the CRT is to apply more stringent image quality criteria when selecting flat-panel technology.

3.2.1.1 Character Size

Character size is an important variable affecting performance error rates. Height and width of the character and the size of the pixel matrix have important effects on human performance. Exercise special care when determining the character size to use on a flat-panel display.

- a. To improve text search and sorting task performance, use a 9 x 13 pixel matrix or larger.

- b. When displaying dot matrix symbols in non-vertical orientations, use at least an 8 x 11 pixel matrix and preferably a 15 x 21 matrix size.
- c. Character Stroke Width (SW) should be in the range defined by: $(\text{character height} + 12) + 0.5 \leq \text{SW} \leq (\text{character height} + 6)$. See Table 3.1 for guidance.

TABLE 3.1: Required Pixels for Stroke Width

Pixels in Upper Case Character Height	Minimum Stroke Pixel Count	Maximum Stroke Pixel Count
7 to 8	1	1
9 to 12	1	2
13 to 14	2	2
15 to 20	2	3
21 to 23	2	4

- d. Character height to width should be in the range defined by: $(\text{character height} \times 0.5) \leq \text{character width} \leq (\text{character height} \times 0.9)$. See Table 3.2 for guidance.

3.2.1.2 Luminance Nonuniformity

Display luminance should be uniform across the surface of the display. Maximum luminance nonuniformity levels should be consistent with the values specified in Table 3.3.

3.2.1.3 Image Formation Time

Image Formation Time (IFT) is the time required to render a new image. Four classes of IFTs (see Table 3.4) have been defined, each relating to information update requirements for the application. IFTs for all systems should be consistent with Classes III and IV.

TABLE 3.2: Required Pixels for Width Design

Pixels in Upper Case Character Height	Minimum Width Pixel Count	Suggested Minimum Width Pixel Count	Maximum Width Pixel Count
7	4	5	5
8	4	6	7
9	5	6	8
10	5	7	9
11	6	8	10
12	6	9	11
13	6	9	12
14	7	10	13
15 or 16	8	11	14

Table 3.3: Maximum Luminance Nonuniformity

Test Object Separation At the Design Viewing Distance	$\frac{L_{\text{higher}}}{L_{\text{lower}}}$	Maximum
$>7^\circ$		1,7
5 to 7°		1,6
4 to 5°		1,5
2 to 4°		1,4
$\leq 2^\circ$		1,3

TABLE 3.4: Image Formation Time Classes

Class	Image Formation Time in Seconds	Significance
I	$120 < t$	Satisfactory for displays that update an entire page of information at once. Noticeable during key entry. Applications using scrolling, animation, and pointing devices are significantly degraded.
II	$55 < t \leq 120$	Satisfactory for displays that update an entire page of information at once. Not noticeable during key entry. Applications using scrolling, animation, and pointing devices are somewhat degraded.
III	$10 < t \leq 55$	Satisfactory for most applications. Motion artifacts can be distracting but are usually acceptable.
IV	$3 < t \leq 10$	Motion artifacts become less noticeable at formation times approaching 3 milliseconds.

3.2.1.4 Display Failures

The three most common failures on matrix-addressable displays are cell failures involving individual elements, vertical line failures, and horizontal line failures. Displays can fail actively or passively and leave pixels or lines permanently on or off, respectively.

- a. Because cell failures often lead to greater performance problems, select displays that minimize the likelihood of cell failure.
- b. To minimize the performance impact of cell failures, select displays and set display polarity so these failures are likely to match the display background.
- c. When display element failure is an expected problem, increase the redundancy in the text to minimize the impact on reading performance associated with display element failures.
- d. Recognition and identification performance with cartographic display is subject to significant decline with as little as 1% pixel failure. Select and maintain displays to ensure a pixel failure incidence below this level.

- e. Use characters with a pixel matrix larger than 7 x 9 pixels in order to reduce the negative effect of "on" failures.

3.2.1.5 Polarity

A display with white (or light) characters on a black (or dark) background is said to have "negative contrast" or to be a "positive (image) display." Conversely, dark characters on a light background are said to have "positive contrast" or to be a "negative (image) display." If character stroke width, modulation, and luminance values are nearly equal for both polarities, select a positive contrast/negative image display for better reading speed, search time, and search error-rate performance.

3.2.2 LIQUID CRYSTAL DISPLAYS (LCDs)

LCDs are perhaps the most developed and popular of flat-panel display technologies. Rather than emit light, as do active flat-panel technologies, an LCD controls or modifies the passage of externally generated light. An LCD is typically made of transparent plate electrodes that sandwich a liquid crystal substance. Voltages applied to these electrodes cause realignment of the liquid crystal material, changing its optical properties and allowing light to propagate through the material. By selectively applying voltage to the electrodes, individual display elements can be made light or dark to create the desired image on the LCD.

The LCD is available in a large variety of formats for both commercial and military applications. Display size and resolution range from small, character-based displays (e.g., those found in watches) to computer displays with up to 640 x 480 resolution. LCDs are monochrome or color and will operate across a wide range of ambient illuminances with backlighting. LCDs are especially suited for information display in environments where ambient illuminances are high.

Advantages of the LCD include excellent contrast, long life, rugged design, low voltage, and low power consumption (except when backlit). LCD technology is limited by slow speed, limited color capability, temperature range, and manufacturing problems for larger panels with higher resolution.

3.2.2.1 Ambient Illumination

Provide adequate levels of ambient illumination, because reading performance improves as ambient illumination increases over the range 20-1500 lux.

- a. Consider LCDs for effective display in high ambient illumination situations.
- b. In low light situations, provide the ability to adjust the viewing angle and the amount of backlight to enhance the legibility of presented information.

3.2.2.2 Polarity

For legibility of transmissive or backlit LCDs, use dark characters on a light background (positive contrast/negative image displays). For reflective LCDs, use light characters on a dark background for better performance.

3.2.2.3 Level of Backlighting

Minimize or eliminate use of backlighting because display reading errors increase as the level of LCD backlighting increases over the range 0-122 candela (cd)/m².

Candela (cd) per square meter (cd/m²) or nit (normalized intensity) is a metric unit for reflected light. One cd or nit is equal to 0.29 footLambert (fL) or one fL is equal to 3.4 nit. A footLambert (fL) is a unit used to measure light reflected from a surface. An ideal surface that reflects all light striking it and diffuses it with perfect uniformity has a luminance of one fL when illuminated by a one footcandle source.

3.2.2.4 Backlighting and Angle of View

Carefully consider the potential impact of user performance decrements before using a backlit LCD that is to be viewed off-axis. There is an adverse impact of backlighting on user performance when the display is viewed at an angle.

3.2.3 GAS PLASMA DISPLAYS

Plasma panels or gas discharge displays are a widely used flat-panel technology in the information and computer industry because of their inherently high-contrast and high-resolution capabilities. Images are formed by ionizing a gas, usually neon, trapped between a set of horizontal and vertical electrodes.

When an electrical field created by the electrodes is increased rapidly, the gas begins to discharge, resulting in a glow that forms an image. The image can be maintained by sustaining the electrical field, or erased by dropping the voltage below some threshold value. The high contrast exhibited by plasma panels is a result of almost no light output in the "off" state (the electrical field is below threshold) and high luminance in the "on" state. Full-color plasma can be made by depositing phosphors on the glass display surface. The plasma gas discharge in this case excites the phosphor, in much the same manner the electron beam does in a CRT, and color images are produced.

Plasma panels can be found as either monochrome or full-color displays in a number of sizes and configurations. A major advantage of plasma technology is that very bright, high-resolution panels are available. Panels that measure 2048 x 2048 pixels at 100 pixels/inch are available, as well as those that can be viewed in direct sunlight. Panels with luminances of 150-600 cd/m² have been produced, with typical large-area, high-resolution display luminances being 30-50 cd/m². Full-color DC plasma panels are not yet able to achieve the luminance output nor the display life normally associated with plasma technology.

Features of plasma technology generally include uniformity, high resolution, large size, long life, ruggedness, and the absence of flicker. Applying plasma technology in the computer and information industry is limited by high voltage and power requirements, complexity of the drive circuitry, low luminous efficiency, need to develop more fully a color capability, and lack of developers of the technology. The literature review conducted for the *Style Guide* determined that specific interface performance information was unavailable for plasma displays. In the absence of specific guidance, the designer should use the most conservative approach to interface design.

3.2.3.1 User Concurrence

Verify the use of gas plasma displays by user personnel as a viable alternative.

3.2.3.2 Testing Prototypes

Designers and developers of computer-based systems should consider field testing prototypes before committing to gas plasma technology.

3.2.4 ELECTROLUMINESCENCE (EL) DISPLAYS

Electroluminescence (EL) displays consist of a layer of polycrystalline phosphor powder or evaporated film phosphors sandwiched between sets of vertical and nearly transparent horizontal electrodes. When an electric field is applied across the polycrystalline phosphor, it is stimulated and light is emitted. The display resolution and the shape of the pixel are defined by the arrangement of the electrodes. EL displays are usually classified as one of four types: either AC or DC thin-film displays, or AC or DC powder displays.

AC thin-film displays and DC powder displays are the most advanced of the EL technologies, and discussions of the strengths and weaknesses of EL technologies will be limited to these types. AC thin-film EL displays have very good luminous efficiency, high contrast, good resolution, and long life. As with some of the plasma technology, these displays require high voltages and complex drive electronics, are expensive, and need research and development to deliver full-color performance. Currently, ELs that display up to 864 lines by 1024 pixels are available for alphanumeric and graphics applications. DC powder ELs have a good appearance, simple structure, good luminous efficiency, and the ability to produce grey scales. However, they require high voltages and complex drive circuitry, and have limited luminance output coupled with high reflectance, which may lead to contrast problems. Also, they are expensive. Full-color displays have been produced using DC powder technology, but this area too needs more development. Currently, resolutions compatible with graphics interfaces for personal computers are available (480 lines by 640 pixels) for use in applications that require alphanumerics and moderate graphics. The greatest problems with EL display technology are that developers are few, and investment in research and development is small.

Interface performance information on EL displays is not available in the current literature. In using EL displays, follow the guidelines recommended for LCDs, and use the most conservative level of those recommendations.

3.2.4.1 User Concurrence

The use of EL displays should be verified by the user personnel as a viable alternative.

3.2.4.2 Demonstration of Concept

The use of EL displays should be field-prototyped before incorporating into a new system.

3.2.5 GLARE-REDUCING TECHNIQUES

Glare, as observed on the face of an electronic display, is composed of two components: 1) Diffuse glare or veiling glare, caused by the general illuminance in the environment, can be characterized as a field effect and has little or no modulation. The effect of diffuse glare is to reduce the effective contrast of the display. 2) Modulated or specular glare is the first surface reflection off the faceplate of the display and results from some object or objects in the area surrounding the display. The effect of this type of glare is the appearance of unwanted images on the display surface, making the displayed information more difficult to see and interpret.

The most effective control of glare is to design appropriate workspace illumination so neither diffuse nor specular glare is produced. This is the only method of glare control that will not compromise the resolution and contrast of the display. Because it is not always possible to control the sources of illumination properly, glare reduction techniques have been developed to minimize the unwanted effects of glare.

Many kinds of glare control techniques are used in the electronic display market. Some are screen meshes placed over the display surface, chemical or mechanical etches of the faceplate of the display, anti-reflective coatings, and bonded quarterwave filters. Each has advantages and disadvantages in terms of ability to control diffuse or specular glare, and in terms of effect on display resolution, flexibility, maintenance, and cost. For example, a bonded quarter wave filter only minimally degrades display resolution but is very expensive, whereas a mesh overlay is very inexpensive but has a major effect on the resolution of the display.

The effectiveness of the glare reduction technique is a function of its ability to suppress each component of glare, while minimizing degradation to the

display's resolution and contrast. The desired effect is to match as closely as possible the display performance under optimum conditions. While both contrast and resolution are degraded in an absolute sense, the effective image quality in the operational environment and the acceptance of the display system should improve.

Selecting the glare control alternative most effective for a particular display depends on the information to be displayed, task required of the operator, and environment in which the display will be used. In a command and control facility, use careful analysis and testing to determine the type of glare reduction measures that should be taken.

3.2.5.1 Reflected Glare

When possible, avoid reflected glare by altering the angular relationship among the observer, display, and glare source. For example, provide the ability to adjust height, viewing angle, and/or contrast.

3.2.5.2 Filter Selection

When possible, leave selection of the specific glare-reduction technique to individual users.

3.2.5.3 First-Surface Specular Reflections

Because many types of display consist of multiple plates of glass, each of these acts as a specular reflector. All flat-panel displays should incorporate a first-surface treatment to diminish first-surface specular reflections.

3.2.5.4 Etched Filters

Etches with gloss values of 45 or less should not be used on monochrome CRTs, and etched filters should not be used at all with high-resolution displays.

3.2.5.5 Projection Displays

With projection displays, minimize glare potential by positioning projection equipment so the light source is not readily visible to viewers.

3.2.6 LARGE-SCREEN DISPLAYS

The DoD operational environment not only imposes requirements for information display at individual workstations, but also for work areas where many persons must observe and use the information presented on a display. Large-screen displays first appeared in the operational environment as the presentation of mission information on transparent plexiglass overlays. Today, large-screen technology found in the command center is computer-driven, with the ability to present graphic and video information.

Large-screen display of surveillance, weather, and intelligence information to operations personnel and as a briefing aid to the principal decision-makers is typical in the operational environment. Unfortunately, because of minimal brightness, poorer contrast, and lower resolution when compared to higher resolution desktop displays, most implementations of large-screen technology have been disappointing to the users. Current display technology offers the military a number of choices when implementing a large-screen display. Figure 3.2 illustrates the types of technology available for fielding large-screen displays. The military currently uses both direct view (e.g., high luminance CRTs or large plasma displays) or projection (e.g., light valves or projection CRTs) large-screen displays in its operational facilities.

Requirements for large-screen display selection in the operational facility vary relative to use and size of the room in which the display will be used. Large-screen displays found in small briefing rooms (approximately 600 square feet) have a screen size of about 50 square feet, horizontal resolution of 300-1000 pixels, and a luminance output of 300-500 lumens. The briefing room must be nearly dark (<2 fc) when the large-screen display is in use.

The other type of facility where large-screen displays may be found is in the command and control center. The command center may be a two-story structure larger than 2400 square feet, where the information is presented to 10-100 personnel. Large-screen displays in this environment have a screen size of about 100 square feet, screen luminance of 10 fL, horizontal resolution of 800-1000 pixels, and a luminance output in the range of 1000 lumens. Room illuminance in the command center is adjustable from about 5-15 fc, but a more normal office illuminance of 75 fc is often requested. Full-color capability is required of large-screen displays in both facilities. Large-screen displays that are larger, brighter, and have more resolution are desired for the command and control environment.

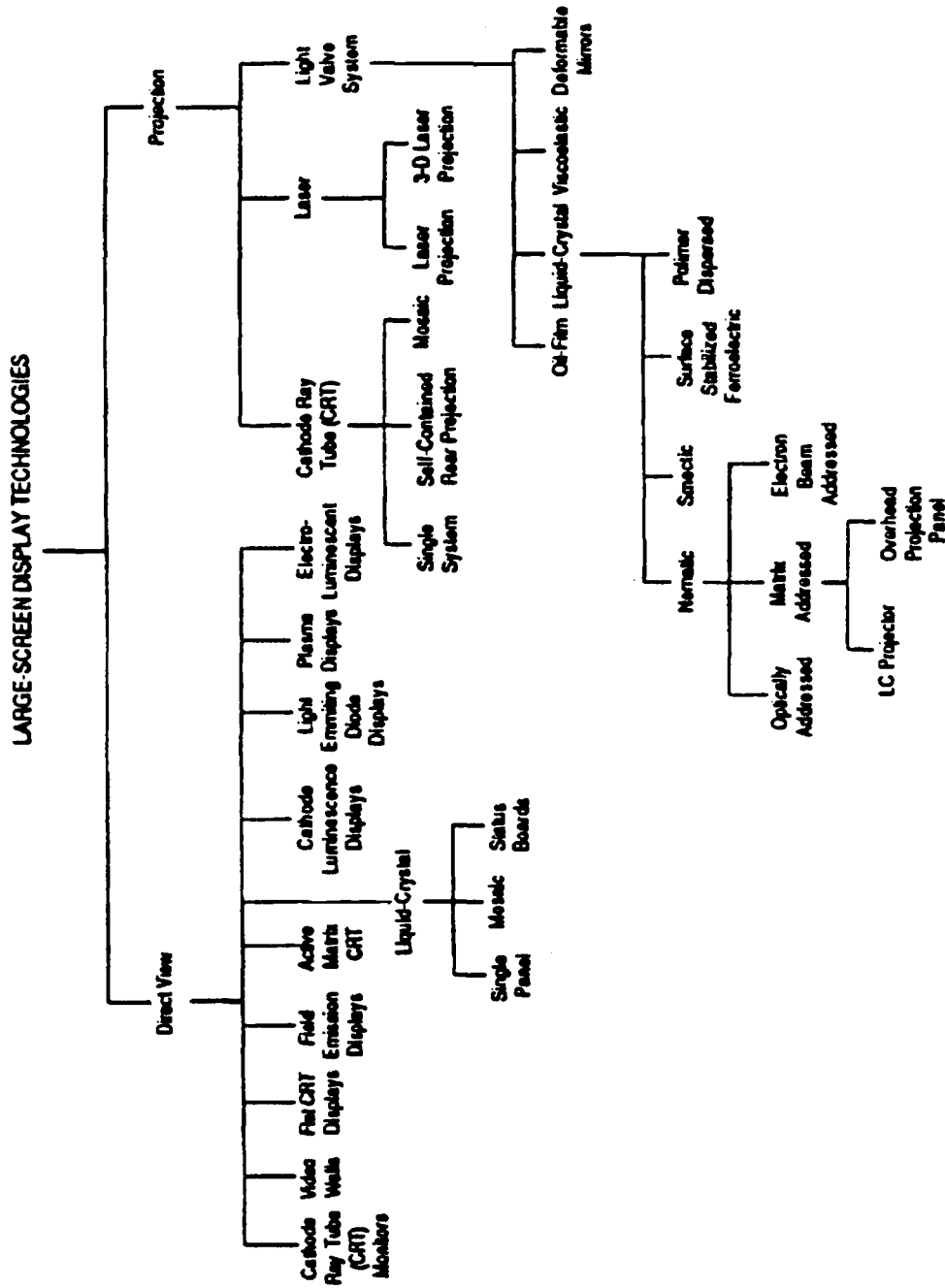


FIGURE 3.2: Large-Screen Display Technologies

Selecting or designing a large-screen display, especially a projection display, may be more complex than for other workstations. The effects of ambient illuminance, observer location, and type of data to be displayed are critical in implementing large-screen display technology. Presentation requirements for data not only relate to one's visual acuity for symbol size and contrast when dealing with projection technology but also to screen size, screen format, symbol luminance, and screen gain. For example, as screen gain increases, the ability to view the screen off the center line decreases. However, a certain amount of screen gain may be necessary to present an image of the necessary contrast, given the expected ambient illuminance in the room. In addition, symbols, graphics, and text should be designed to compensate for the degraded viewing conditions that may exist in the operational environment due to a number of factors. Implementing large-screen displays in the operational environment should always take into account the environmental factors, as well as the information display requirements.

Typical office ambient level is greater than 75 fc, whereas typical command and control centers are 5-15 fc.

3.2.6.1 Character Dimensions

Because information is often viewed off the center axis, use character sizes between 10 and 20 minutes of visual arc and with a minimum of a 10 x 14 dot matrix format.

3.2.6.2 Stroke Width

The ratio of character stroke width to character height should be 1:6 to 1:10. Characters with double stroke widths should be used in situations requiring off-axis, longer distance, and/or viewing under difficult lighting conditions.

3.2.6.3 Luminance

Modulated output luminance, spatially averaged over the full screen, should be 300-400 lumens for small conference rooms and command posts and 750-2000 lumens for a control center, assuming 20-40 footcandles (fc) ambient lighting in each case.

3.2.6.4 Size versus Luminance

To ensure legibility, small characters (5 min. arc) require contrast ratios of 15-20:1, whereas large characters (>20 min. arc) require contrast ratios of 1.5-5:1.

3.2.6.5 Aspect Ratio

Aspect is the ratio of the horizontal to vertical dimensions of a character or image. Character aspect ratio should be approximately 1.33:1.48 (width:height ratio).

3.2.6.6 Modulation Depth

A display should deliver at least 15% visual contrast when measured as modulation depth $[(L_{max}-L_{min})/L_{max}]$, when an alternating pixel pattern is displayed at normal luminance levels.

- a. The contrast ratio between the reflected luminance of the screen with a projected light source and the reflected luminance of the screen without a projected light source should be approximately 500:1.
- b. Negative contrast (black characters on a white background) should be used.

3.2.6.7 Displayed Data Characteristics

- a. Avoid displaying too much data. As with standard displays, consider data type, amount, and appropriate sequence of presentation in designing large-screen display screens.
- b. If displaying color-coded targets, use only a neutral color such as grey for the background.

3.2.6.8 Projection Equipment

- a. Minimize glare potential by positioning projection equipment so it is not readily visible to viewers.
- b. To minimize optical distortions, ensure image source equipment and the projection screen are fully parallel. Electronic or optical distortion-compensating devices may be used to compensate for any remaining distortion and to assure clarity of displayed information.

- c. Consider using rear projection or other direct view large-screen displays when increased contrast demands are encountered and/or when there is a need to position personnel in the field of projection.

3.2.7 STEREOSCOPIC/3D DISPLAYS

The display of three-dimensional (3D) images and graphics is an emerging technology that may benefit future military applications. Examples of where 3D technology may be used are in battlefield and theater of operations analysis, photo-interpretation, teleoperation, air space control, engineering, and training and simulation exercises. The goal of introducing 3D displays is to improve user performance and increase naturalness of the interaction. Currently, most 3D technology is experimental and, as such, is not suitable for an operational environment, although a few stereographic and true 3D electronic displays can be purchased commercially.

Because traditional display technology is a two-dimensional medium, it has not been able to take full advantage of the human visual system to interpret complex spatial data. Binocular depth information, such as vergence or horizontal disparity, normally in the scene, are not available in the traditional electronic display. Compensation for this has been accomplished by using monocular cues, such as interposition, shading, and perspective. However, improvements in naturalness of the display and potential for gains in human performance with computing systems have stimulated the development of systems that make use of the stereoscopic capabilities of the human visual system.

Three-dimensional display technology is classified as stereoscopic and autostereoscopic. The major criterion distinguishing stereoscopic displays from autostereoscopic displays is that the latter requires no special viewing aids to see the 3D image. There is also a difference in the amount of information necessary to create the 3D image.

Stereoscopic displays create the 3D image by requiring an observer to wear a pair of glasses that provides separate images to the left and right eye. When the alternate fields are presented to the eye sequentially at the appropriate rate, the illusion of depth is created. The temporal phase difference that accounts for the stereopsis creating the 3D illusion has been implemented normally by requiring the viewer to wear a pair of glasses containing either shuttered lenses, polarized lenses, or red and green lenses. By synchronizing the image presentation to the operation of the glasses, images corresponding to the left

and right scenes are presented to the viewer and the illusion of depth is created.

Autostereoscopic displays, by contrast, can be viewed directly. These displays generally use a multiplanar approach to add depth to a two-dimensional image. Examples of this type of 3D display are BBN's SpaceGraph 3D Display System (uses a flexible mirror to provide the z axis), Tectronix liquid crystal shutter 3D display (uses LCD technology together with a CRT display to create a 3D effect), and Texas Instruments' Omniview (uses a rotating multi-planar surface to produce the z axis). Holography has also produced 3D images, but none to date has been created in real time.

While innovative technology to provide 3D images is becoming available, no clear guidance outlines where stereoscopic displays might best affect task performance or subjective image quality. Additionally, no database derived from applied vision or human factors research currently exists for developing application guidelines for this new technology. Consequently, the system designer must be cautious in applying 3D display technology in the military environment. Current technology often limits field of view, number of observers, and type of data that can be presented. It also may exacerbate visual deficiencies that normally have little effect on task performance. Guidance presented here is by no means complete. Many questions remain unanswered, both in terms of human visual response to artificially generated depth from electronic displays and the ways best to enhance performance using this technology.

3.2.7.1 Purposeful

Presenting 3D information must be purposeful to benefit the user. That is, 3D displays should be associated with the type of work to be performed and required for task completion.

3.2.7.2 System Performance

Presentation of 3D or depth information should not slow information updates, degrade other aspects of system performance, or degrade image quality.

3.2.7.3 Interocular Crosstalk

Interocular crosstalk or bleed-through occurs in stereoscopic displays when images intended for the left eye are seen by the right eye and vice versa.

Because this compromises the observer's ability to fuse the image and perceive it as a 3D object, there should be zero interocular crosstalk between the two images.

3.2.7.4 Color Coding

Avoid saturated primary colors, as these colors may evoke depth perceptions that may be inconsistent with stereopsis, affecting the perception of depth. Designers should use secondary colors rather than saturated primary colors in coding stereoscopic images.

3.2.7.5 Symbols

When displaying symbols, disparity should range from 0 to 20 minutes of arc in both crossed and uncrossed directions.

3.2.7.6 Dynamic Depth Displays

When using dynamic depth displays, the temporal modulation of stereopsis should be approximately 1 Hz to ensure the most accurate perception of stereomotion.

3.2.7.7 Depth-Coded Objects

Spatially separate depth-coded objects in stereoscopic images to eliminate disparity averaging, crowding, or repulsion.

3.2.7.8 Size Scaling

Image size should be scaled according to the disparity of the image. When accurate size perception is critical to task performance, size scaling should be done for an individual observer.

3.2.7.9 Luminance

Because brightness is also a depth cue (bright objects are viewed as nearer), luminance should be co-modulated with stereopsis.

Display parallax is the apparent displacement of an object displayed on a curved CRT screen and viewed through a flat Touch Interface Device.

3.2.8 TOUCH INTERACTIVE DEVICES

A Touch Interactive Device (TID) is an input device that permits a user to interact with the system by pointing to objects on the display. The TID is addressed in this section, because some implementations of touch technology can severely degrade the quality of the displayed image. Degradation in image quality using TIDs may result from decreases in display luminance, reductions in display resolution due to visibility of conductors or the device material, increased susceptibility to glare, and dirt on the display surface resulting from touching the display surface. Display parallax, caused by separation between the touch surface and the touch targets, may also contribute to problems with implementing TIDs.

There are six basic types of touch-screen display technologies, each having an impact on display parallax, transmissivity of light, and glare. Each is briefly discussed below. The designer needs to be aware of the advantages and disadvantages of each type of TID when selecting hardware and designing interfaces using TIDs.

- a. Fixed-wire TIDs place wires, either in parallel or in grid fashion, in front of the display. Finger contact with the wire(s) signifies the x,y coordinate of the user's response. This technology is associated with minimal parallax, 70-80% transmissivity, and a medium to high degree of TID glare.
- b. Capacitive TIDs consist of a transparent conductive film on a glass overlay. Touching this surface changes the small electrical signal passing through the surface, and this signal is converted into the corresponding x,y coordinate. This technology is associated with minimal parallax, 85% transmissivity, and a medium degree of TID glare.
- c. Resistive membrane TIDs are "sandwich" devices in which a touch results in the contact of two conductive layers. Specific current and voltage levels are associated with individual x,y coordinates. This technology is associated with minimal parallax, 50-60% transmissivity, and a high degree of TID glare.
- d. Infrared (IR) or light-emitting diode TIDs use IR transmitters along two perpendicular sides of the display frame and photocell receptors along the opposite sides of the frame. A user touch breaks the resulting matrix of light beams, and the appropriate x,y coordinates of the touch are thus determined. This technology is associated with no parallax problems in seeing the display (although a noticeable degree of parallax exists between the plane of the IR

grid and the screen surface for touch responses), 100% transmissivity, and no TID-related glare.

- e. Surface acoustic wave TIDs operate in similar fashion to IR TIDs, except that the matrix overlay is one of ultrasonic sound beams rather than IR beams. Another approach, called the reflective array, uses a piezoelectric transmitter and a series of reflectors and receivers. Touch x,y coordinates are determined by differential timings in reception of the acoustic waves. At least some devices require glass overlay screens. This technology is associated with minimal parallax, 92% transmissivity, and a medium degree of TID glare.
- f. Pressure-sensitive devices use strain gauges mounted between the display screen and an overlay. Output voltages of these strain gauges are encoded into the appropriate x,y coordinates. This technology is associated with minimal parallax and zero TID glare. Figures for transmissivity are not applicable because the overlay is built into the display screen.

3.2.8.1 Parallax

TID/display parallax should be minimized because it has been shown to lead consistently to poorer entry time and touch count performance.

3.2.8.2 Specular Glare

Specular glare should be minimized for applications using TIDs.

3.3 ALTERNATE INPUT/OUTPUT (I/O) DEVICES

The focus of HCI design literature and research has been on the software, displays, physical environment, and computer equipment aspects of the interface. The approaches to testing and evaluating HCIs are usually based on the machine rather than the human portion of the computer interface. The perceptual characteristics of the expected user are rarely investigated, and interface design ignores known population perceptual limitations. Using color to transfer information does not take into account the potential of color-deficient vision problems in user populations. Using auditory codes does not take into account potential hearing deficits by frequency and adjust outputs according to known population characteristics. The distribution of visual acuity within the user population is usually not considered. It is more likely that environmental impacts on the system will be defined than will user perceptual characteristics.

The accessibility of computer-based systems by people with disabilities is U.S. Government policy based upon Public Law 99-506 and Public Law 100-542. Individuals with limited hearing, vision, or mobility require enhancements to existing computer-based systems in order to effectively use these resources. These laws address the requirement that the acquisition and management of Federal Information Processing (FIP) resources be conducted in a manner that ensures employees with disabilities access to computer and telecommunications products and services. The implementing regulations for these laws are contained in the Federal Information Resource Management Regulation (FIRMR), 41 CFR Chapter 201.

The interface designer must identify computer and telecommunication accessibility requirements for current and prospective users. The functional aspects of user requirements are an important part of system design and implementation. When automated information environments offer the needed flexibility, users with limitations in vision, hearing, or mobility are ensured full access and integration at a level equivalent to users without disabilities. Flexibility can be achieved in most information environments through off-the-shelf "drop in" or "add on" hardware and software enhancements that modify the common input (e.g., keyboard or mouse) or output (e.g., monitor or printer) interactions associated with computer operations. In addition to being more user-responsive, input capability may need to offer portability, speech input, or wireless connection to the computer. The output may need to be enhanced by magnified text or synthesized speech.

3.3.1 VISUAL I/O

3.3.1.1 Low Vision

The term "low vision" covers a broad range of possible conditions and types of visual impairment. The following techniques will enhance conditions for the visually impaired user.

- a. The use of glare protection technology can minimize visual fatigue associated with glare on a monitor.
- b. The use of monitors from 19 to 25 inches will allow increased character size and provide a larger image display.
- c. Software or hardware can be used to present images on a computer monitor in a large format. Character sizes can be increased.
- d. Software can be used to modify the print size on graphic printers.
- e. The user should be allowed to select color schemes for aspects of the application that do not involve coding or status. A default scheme should be easily available to restore the interface for subsequent users. The control of color will allow the user better contrast control.
- f. The keyboard can be tactilely coded with raised dot or bleb. Keycap labels with larger letters can be added.

3.3.1.2 Blind

The user with very limited or no usable vision will require additional interface enhancement.

- a. The use of speech recognition technology for input and speech synthesizers for output will provide an interface for the visually impaired.
- b. Hardware and software are available to convert standard word-processing documents so they can be printed on a braille printer. Dynamically displayed braille is available on output devices. Braille note-takers' devices are available that are capable of braille input and output to a PC.

- c. The use of an OCR (Optical Character Reader) device can translate printed material to speech or braille formats.

3.3.2 HEARING I/O

3.3.2.1 Visual Redundancy

Ensure that information conveyed by beeps or speech during computer-related tasks are also displayed visually for the user unable to benefit from the auditory information.

3.3.2.2 Amplification

Ensure that auditory output from the computer interface has adjustable volume and frequency range.

3.3.2.3 Signalling

Ensure that alerts and others signals related to the interface are presented in another modality, such as tactile or visual.

3.3.3 MOBILITY

A variety of interface solutions are available for users with various degrees of limited mobility. In addition to the computer interface, review the entire work environment for barriers to access.

3.3.3.1 Keystroke Input

The interface hardware and/or software can be modified to allow for sequential keystrokes rather than simultaneous. Keyboard macros can be created to reduce the number of keystrokes required. The repeat rate of keys can be adjusted and the pressure required to activate keys can be modified to the user requirements.

3.3.3.2 Keyboards

Alternate keyboards are available that may be more easily used by mobility-impaired individuals. There are also devices that replace keyboards, such as muscle switches, optical pointers, sip and puff systems.

3.3.3.3 Speech I/O

The use of speech recognition technology for input and speech synthesizers for output will provide an interface for the mobility-impaired user.

3.3.3.4 Pointing Devices

The interface should not be dependant on pointing devices. The interface should have redundant input/output capability through the keyboard. The selection of pointing device should consider the mobility parameters of the intended user.

3.3.3.5 Optical Character Recognition (OCR)

The use of an OCR device can allow the translation of printed material to a speech-compatible interface.

REFERENCE LIST

<u>Paragraph</u>	<u>References</u>
3.1	DISA/CIM (1992)
3.2	Benson and Farrell (1988); Lloyd et al. (1991); Dye and Snyder (1991); Decker et al. (1991); Reger et al. (1989); Biberman and Tsou (1991)
3.2.1	Tannas (1985) page 11; Goode (1991); Beaton and Knox (1987)
3.2.1.1	Reger et al. (1989); Petrun et al. (1985)
3.2.1.1.1	Laycock (1985)
3.2.1.1.2	DoD (1989a) par. 5.2.6.8.3
3.2.1.1.3	ISO (1991) Part 3 Addendum
3.2.1.1.4	ISO (1991) Part 3 Addendum
3.2.1.2	ISO (1991) Part 3 Addendum
3.2.1.3	ISO (1991) Part 3 Addendum
3.2.1.4	Lloyd et al. (1991)
3.2.1.4.1	Dye and Snyder (1991); Lloyd et al. (1991)
3.2.1.4.2	Lloyd et al. (1991); Laycock (1985)
3.2.1.4.3	Lloyd et al. (1991)
3.2.1.4.4	Dye and Snyder (1991); Lloyd et al. (1991)
3.2.1.4.5	Lloyd et al. (1991)
3.2.1.4.6	Dye and Snyder (1991); Lloyd et al. (1991); Snyder (1988); McNeese and Katz (1986)
3.2.2	Snyder (1980); Goode (1991)
3.2.2.1	Petrun et al. (1985); Payne (1983)
3.2.2.1.1	Payne (1983); Muracka et al. (1989)
3.2.2.1.2	Biberman and Tsou (1991), Reger et al. (1989); Cristensen et al. (1985)
3.2.2.2	Kubota et al. (1988), Kubota Murushige Takabayashi and Kobayashi (1986)
3.2.2.3	Payne (1983)
3.2.2.4	Payne (1983) Muracka et al. (1989)
3.2.3	Tannas (1985) Chapter 10; Goode (1991); Snyder (1980)
3.2.4	Snyder (1980); Goode (1991)
3.2.5.1	Reger et al. (1989); Petrun et al.(1985)
3.2.5.2	Morse (1985)

REFERENCE LIST (Cont.)

<u>Paragraph</u>	<u>References</u>
3.2.5.3	ISO (1991) Part 3
3.2.5.4	Hunter (1988)
3.2.5.5	Woodson et al. (1992)
3.2.6	Breen (1987); Breen (1989); Hurley and Hoffberg (1991); Carbone and MacIver (1987)
3.2.6.1	McNeese and Katz (1986); Shurtleff et al. (1982)
3.2.6.2	Woodson et al. (1992)
3.2.6.3	Blaaha (1990); Breen (1989)
3.2.6.4	Shurtleff et al. (1982)
3.2.6.5	Woodson et al. (1992)
3.2.6.6	Blaaha (1990)
3.2.6.6.2	McNeese and Katz (1986)
3.2.6.7.1	Woodson et al. (1992)
3.2.6.7.2	Woodson et al. (1992)
3.2.6.8.1	Woodson et al. (1992)
3.2.6.8.2	Woodson et al. (1992); Vance (1987)
3.2.6.8.3	Woodson et al. (1992)
3.2.7	Veron et al. (1990); Yeh and Silverstein (1989); Reinhart (1990); Beaton (1990); Snyder (1984); Williams (1990)
3.2.7.1	Beaton (1990)
3.2.7.2	Beaton (1990)
3.2.7.3	Yeh and Silverstein (1991)
3.2.7.4	Yeh and Silverstein (1991)
3.2.7.5	Yeh and Silverstein (1989); Yeh and Silverstein (1991)
3.2.7.6	Yeh and Silverstein (1991)
3.2.7.6	Yeh and Silverstein (1991)
3.2.7.7	Yeh and Silverstein (1991)
3.2.7.8	Yeh and Silverstein (1991)
3.2.7.9	Yeh and Silverstein (1991)
3.2.8	Dominessy (1989)
3.2.8.1	Baggen et al. (1988)
3.2.8.2	Baggen et al. (1988)
3.3	GSA (1991)

4.0 SCREEN DESIGN

Screen design refers to how information is arranged and presented on a display screen. It is difficult to develop standard guidelines for screen design for the range of DoD systems, primarily because of the differences in tasks being performed by users. Screen design requirements can vary extensively, depending on the function being performed by the system. Some systems are actually information management systems that rely heavily on databases. These systems do not require immediate user response to information displayed on their screens. On the other hand, real-time tactical display and control systems require the user to make immediate decisions and to input commands from the information on the display screen. Each system has different screen design requirements based on its primary function. The designer needs to understand the primary function of the system being developed to provide an effective screen design.

Incorporate the following general principles of Human Factors Engineering (HFE) design into the screen design, regardless of the system function:

- Guide the organization of information by Gestalt principles of perception, such as rules of proximity and similarity.
- Design display formats to provide optimum transfer of information to the user by the use of information coding, density, grouping, and enumerating.
- Present information simply and in a well-organized manner.
- The user's performance is improved by the following screen features:
 - an orderly, clutter-free appearance
 - information present in expected locations
 - plain, simple language
 - a simple way to move through the system
 - a clear indication of interrelationships

- Displays should be formatted to group data items on the basis of some logical principle, considering trade-offs derived from task analysis.
- Screen design should minimize pointer and eye movement requirements within the overall design. The goal to minimize eye and pointer movement must be considered within general task considerations, with logical trade-offs taken into account.

The following guidelines represent design considerations that should be applied to the screen design of all DoD systems.

4.1 INITIAL SCREEN DESIGN

This section provides guidelines for log-in, log-off, the initial screen display, and the management of workstation resources. The examples are drawn from intelligence application, but apply to all system designs. The specific security recommendations may or may not apply to any given application.

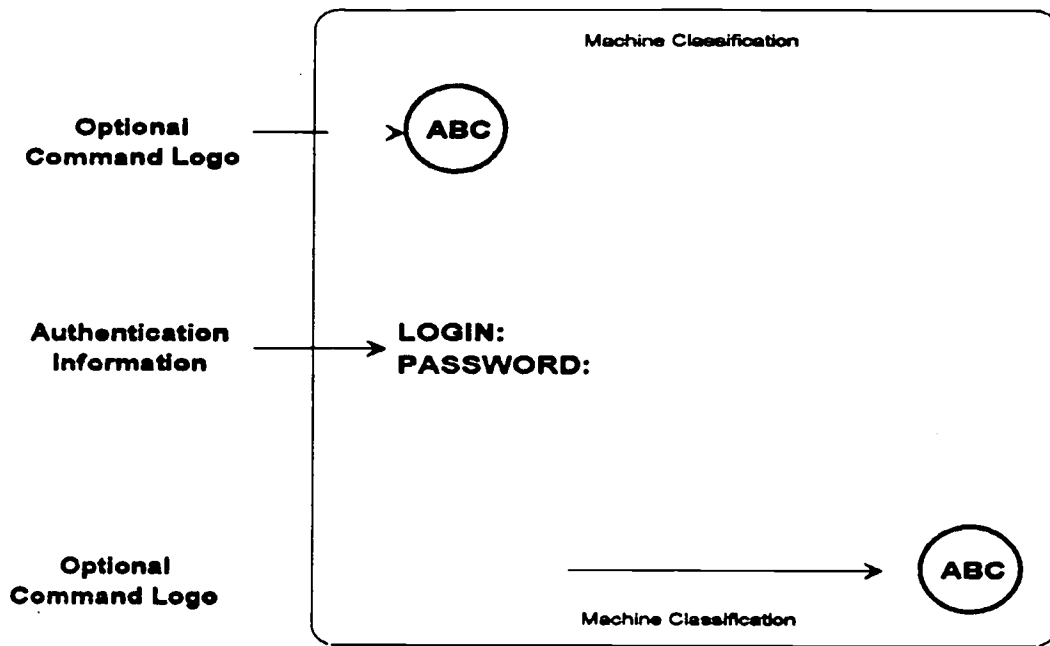
4.1.1 WORKSTATION LOG-IN

A standard workstation log-in screen should be developed for each application (see Figure 4.1). Rather than continuous display of the log-in screen or any other display on an idle workstation, all workstations should implement a screen saver. The screen saver should be activated when the workstation has been idle for an adjustable period of time and deactivated whenever any new activity is detected.

Guidelines for developing a workstation log-in procedure are as follows:

- The security authentication information (when required) should be a combination of name, password, and/or other identification information required before a user can access system resources.
- Each prompt for the user's name, password, etc. should be clearly labeled and displayed on a separate line.
- Error messages should be clearly displayed at the bottom of the computer screen along with guidance on how to correct the error. Error messages or help generated during the log-in sequence should not convey information that could assist someone in breaking into the system.

LOG-IN SCREEN



INITIAL WORKSTATION SCREEN

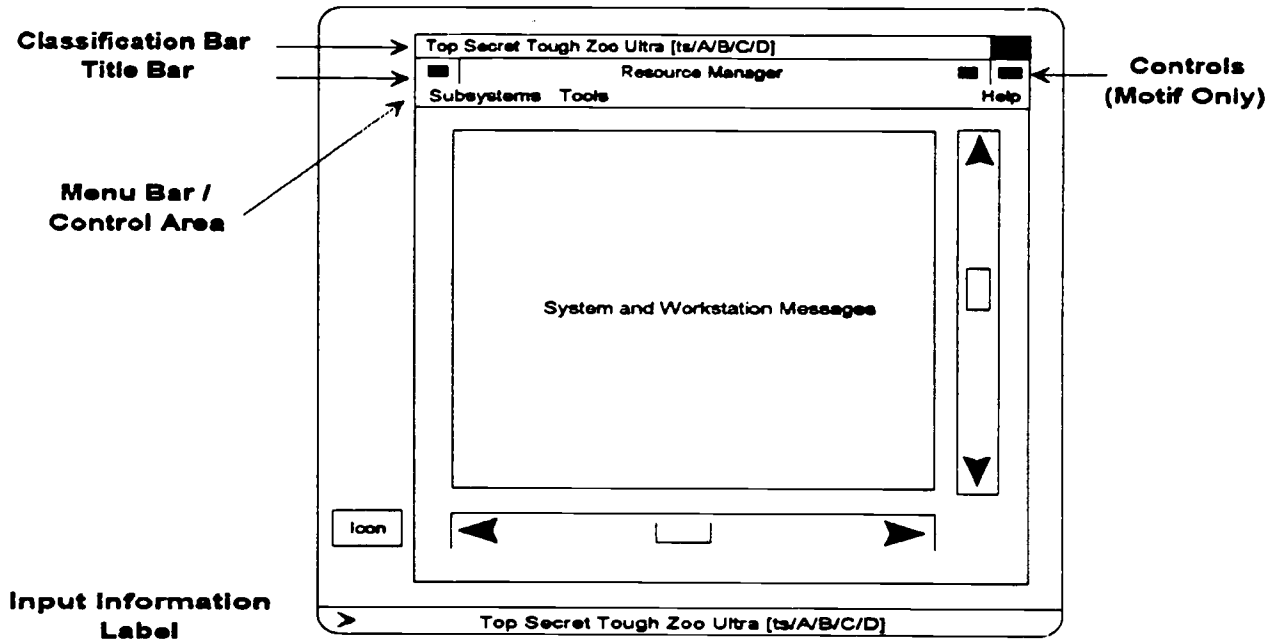


FIGURE 4.1: Sample Workstation Screens

When displaying a machine classification on a workstation accredited for compartmented mode operations, the lowest classification applicable to all possible users of the workstation should be displayed. When displaying a machine classification on a workstation accredited for system high operations, the system high banner should be displayed.

4.1.2 APPLICATION LOG-IN

A primary DoD architectural objective is to implement unitary log-in, but some systems will be unable to support this feature immediately. In systems where unitary log-in is not supported, many applications will require a separate authentication process before they can be accessed. Following selection of such an application by the user, an additional log-in should be displayed to prompt the user for the required authentication information.

4.1.3 APPLICATION LOG-OFF

Application log-off exits an application and closes all windows associated with the application. If there are other open windows, the next window on the window stack should be uncovered; otherwise, the resource manager window (see Section 4.1.5) should appear on the screen. Application log-off is accomplished by selecting the Exit function. If work has not been saved, the user should be asked to confirm the quit, save modified data, or cancel the request.

4.1.4 WORKSTATION LOG-OFF

Workstation log-off ends the session, closes all application windows, and returns the computer screen to the initial workstation log-in screen. If any applications are running, workstation log-off should also initiate an exit from all active applications. Workstation log-off requires user confirmation and is accomplished by selecting a log-off option from the resource manager window.

4.1.5 INITIAL WORKSTATION SCREEN DISPLAY

When the user successfully completes the workstation log-in procedures and has been granted access to system resources, the resource manager window will be displayed on the screen. This section outlines the capabilities of a resource manager and describes the basic computer screen. General window functions are discussed in Section 5.

4.1.5.1 Resource Manager

The resource manager is the function that provides access to workstation resources and utilities (e.g., drives, printer, files, applications, software packages, etc.). This function is often referred to as session management, but to avoid ambiguity, this *Style Guide* discusses the management of workstation resources under the generic term of resource management. The resource manager should provide at least the following capabilities:

- program accesses
- window snapshots (print screen)
- access to common applications (e.g., word processor, spreadsheet)
- user preference/customization (e.g., left or right-handed mouse)
- utilities (e.g., calculator, calendar, clock/alarm, note pad, mail)
- display of system and workstation messages (error and status)
- end session/log user out of account
- work file maintenance
- system-level help
- security functions for authorized persons.

The resource manager should present only those functions and applications a particular user is allowed to access. For example, only users authorized to perform certain security functions should have those options available within a resource manager menu.

4.1.5.2 Resource Manager Menu

The resource manager menu should contain, as a minimum, menus for applications the user is allowed access and help. Figure 4.1 illustrates a sample initial workstation screen.

A long-term DoD objective is to implement functionally oriented (versus system- or application-oriented) resource manager menus. Users may require data from several systems to perform their specific jobs. When multiple data sets must be accessed to satisfy a user query, it is the responsibility of the application to determine where the data resides.

4.2 SCREEN DESIGN GUIDELINES

4.2.1 GENERAL

4.2.1.1 Information Display Based on Criticality

A set of criteria should be established for prioritizing different levels of displayed information. For example, critical tactical information should always be displayed, whereas optional information should be available by request. See Figure 4.2.

4.2.1.2 Display Only Critical Information

The information density on a system display should be minimized by presenting only information essential to the user at the time.

4.2.1.3 Integrated Display

When the user needs specific data displayed concurrently to make judgments on the tactical situation, provide those data in an integrated display, rather than partitioning them into separate windows.

4.2.1.4 Information Format

Present information in a directly usable form. Do not require the user to decode or interpret data. The structure of information presented on the screen should be consistent. This helps the user develop a perceptual model of the interface.

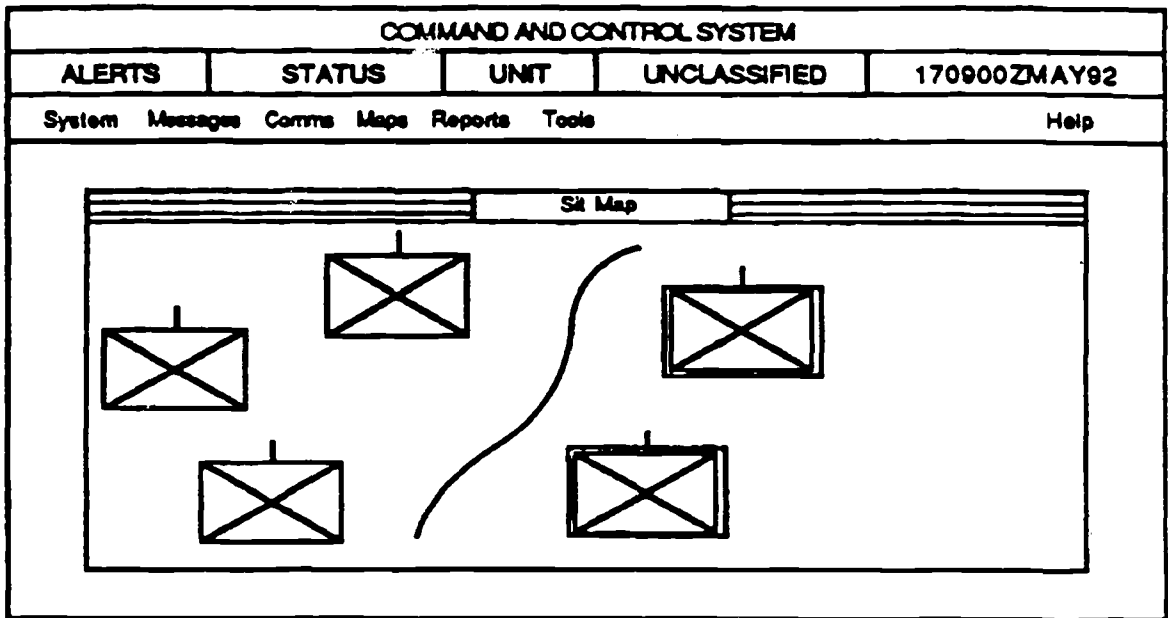
4.2.2 CONSISTENCY

4.2.2.1 Consistent Display Structure

Create display formats with a consistent structure evident to the user, so display features are always presented in the same way.

4.2.2.2 Consistent Fields

Use fields, such as headers, that are constant and stay the same on every page.



Display Screen Design Where Critical Information is Always Present

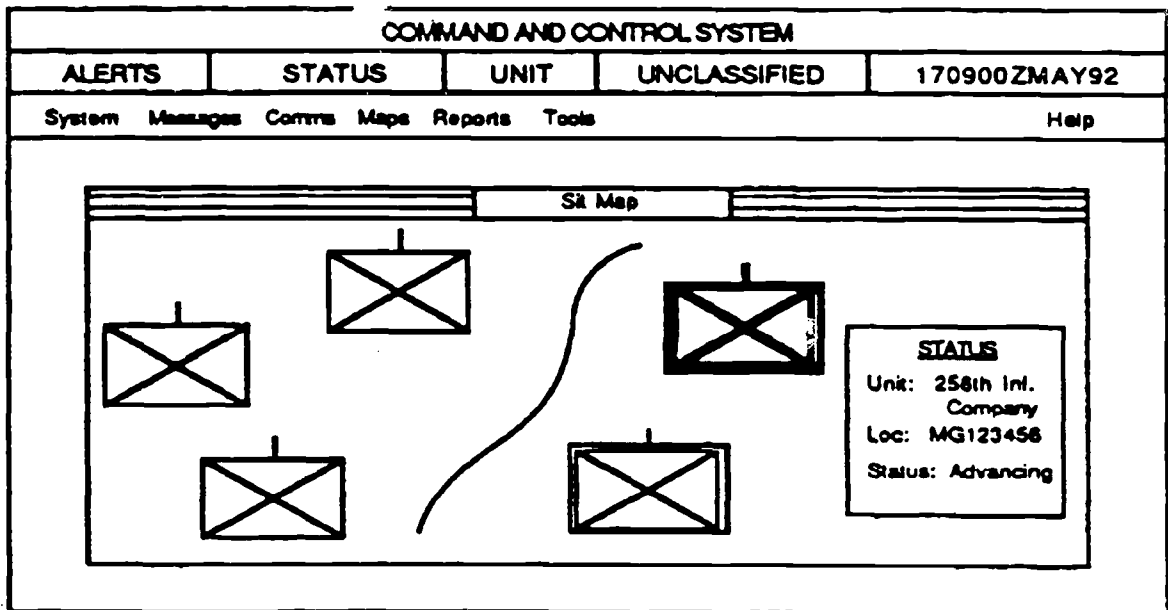


FIGURE 4.2: Tactical Information Display Options

4.2.2.3 Input Prompts

Display the input prompt at a standard location, next to the command entry area of the display.

4.2.3 FORMAT

4.2.3.1 General

- a. Make the different elements in a display format distinctive.
- b. Information on a display screen should be organized such that visual competition among distinct items of information is minimized.
- c. Make appropriate use of abbreviations, and use them in a consistent manner. Provide a key or built-in reference table. Abbreviations should conform to AR310-50 (U.S. Department of the Army 1985a). Do not place periods after abbreviations. It is recommended that applications requiring extensive text input provide an on-line spell-checker that addresses abbreviations and acronyms.
- d. Use contrasting features such as inverse video and color to call attention to different screen components and urgent items.
- e. Use short, simple statements in text.

4.2.3.2 Screen Organization

- a. The order of data should follow some principle that can be recognized and applied by the user.
- b. Begin every display with a title or header located at the top of the page, describing briefly the contents or purpose of the display. Leave at least one blank line between the title and the body of the display. See Figure 4.3 for an example.
- c. The area set aside for displaying messages should be consistent. Text systems have reserved the last several lines at the bottom of displays for status and error messages, prompts, and command entry, when appropriate (Figure 4.3). This area can also be used for a supporting data menu bar, including such items as a user note pad.

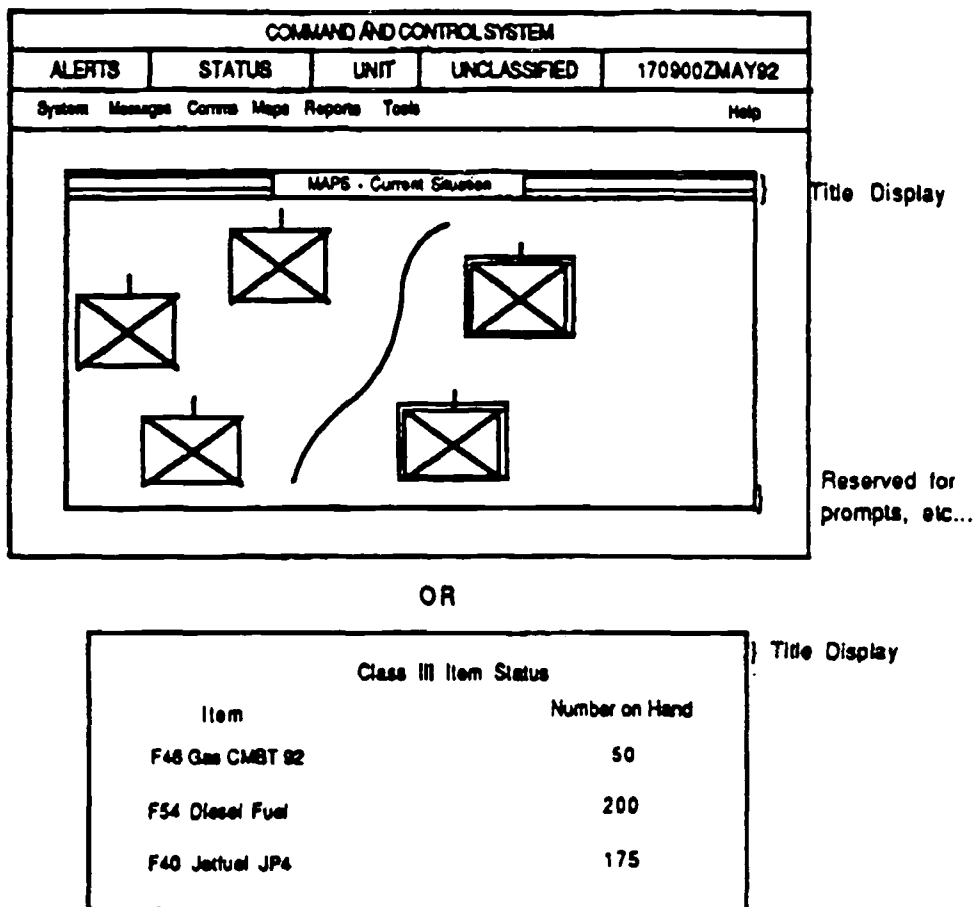


FIGURE 4.3: Example of How Specific Types of Information Should Be Located on a Display Screen

- d. For text displays, screen density (i.e., ratio of characters to blank spaces) should not exceed 60% of available character spaces.
- e. Instructions on how to use a screen should be highlighted at the top of the text, preceding the response options, as illustrated in Figure 4.4. Instructions on the disposition of a completed screen should be at the bottom.

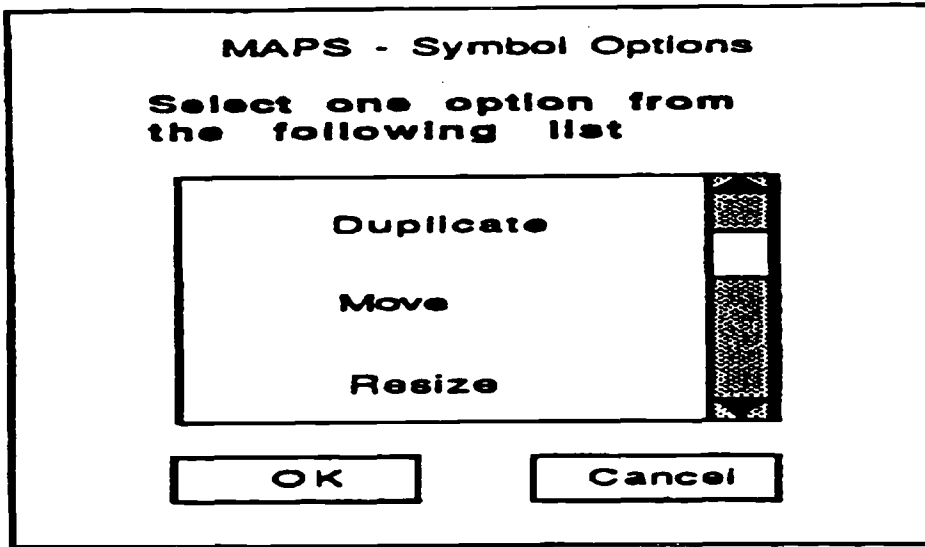


FIGURE 4.4: Example of the Proper Location of Display Screen Instructions

- f. Assign functional fields for particular kinds of data, such as program messages, error messages, system messages, and alarms.

4.2.3.3 Data Organization

- a. Large portions of text should be broken into smaller, meaningful groups to minimize the amount of information to be attended to at one time (Figure 4.5).
- b. Use blank space to structure a display.

<u>Poor</u>	<u>Good</u>
<p>The 3rd Bn is currently located at 32UNA100100, moving to contact in sector 8, with 80% strength, supported by an armor platoon.</p>	<p>3 Bn Status</p> <ul style="list-style-type: none"> o At 32UNA100100 o Moving to Contact o 80% Strength o Supported by an armor platoon

FIGURE 4.5: Example of Breaking Information into Smaller Groups

- c. For screens containing large amounts of text, consider using two columns of text to improve readability.
- d. Ensure labels are sufficiently close to their related data fields but separated by at least one space.
- e. Provide adequate spacing between words and lines of text for better legibility. Separate paragraphs with a blank line.
- f. A series of data elements should be presented vertically, not horizontally, in text (Figure 4.6).
- g. Provide an obvious starting point for information.

<u>POOR</u>			
Class I	Item 1	Item 2	Item 3
Class II	Item 1	Item 2	Item 3
Class III	Item 1	Item 2	Item 3
<u>GOOD</u>			
<u>Class I</u>	<u>Class II</u>	<u>Class III</u>	
Item 1	Item 1	Item 1	
Item 2	Item 2	Item 2	
Item 3	Item 3	Item 3	

FIGURE 4.6: Example of Vertical Data Presentation vs Horizontal

<u>Poor</u>	<u>Good</u>
Artillery Tanks Trucks Aircraft	Artillery Tanks Trucks Aircraft
-----	-----
400 4210 38 39111	400 4210 38 39111
-----	-----
1.5 10.38 1.365	1.5 10.38 1.365

FIGURE 4.7: Example of How Data Should Be Justified

- h. Columns should be justified, as noted below and as illustrated in Figure 4.7.**
 - 1) Left justify alphanumeric columns to permit rapid scanning.**
 - 2) Right justify numerical data without decimals.**
 - 3) Justify numerical data with decimal points by the decimal.**

4.2.3.4 Line Organization

- a. Use only 70-character positions on the standard 80-character line to increase reading efficiency. This is most important when detailed reading is the user's primary task.**
- b. Display no more than 35 to 40 characters on each line for information presented in columns.**

4.2.3.5 Character Design

- a. Capital letters should be used for typographic coding, headlines, indicating new paragraphs, and/or captions and labels. In some cases, such as labels of buttons only the first letter is capitalized.**
- b. Do not use all capital letters in running text or tables. It impairs word recognition, reduces readability, and limits space between text lines.**
- c. Spacing between characters should be 20-50% of character height. Spacing between lines should be equal to character height. See Figure 4.8.**
- d. Minimum height of displayed characters should be 1/200 of viewing distance. For example, a viewing distance of 36 inches requires a 0.18-inch character height on the display screen. Character width should be 50-100% of character height. Character stroke width minimum is 10-12.5%**

of character height. Maximum text size should not exceed 10% of the available vertical display area on a full-size screen.

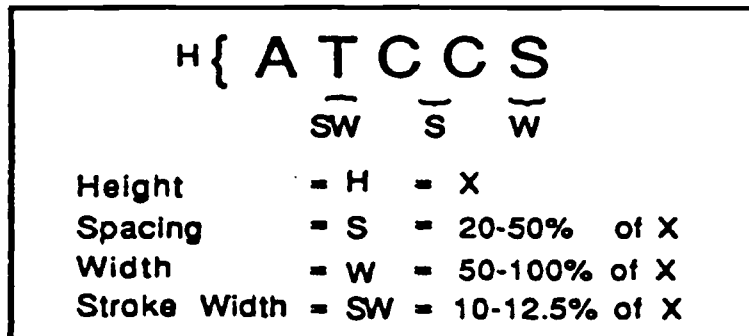


FIGURE 4.8: Example of Character Size and Spacing

- e. Characters should contain a minimum 7 x 9 dot matrix construction for better readability.
- f. Operational applications require the characters on the screen be legible to a person standing behind the user. The screen viewing distance referred to in Paragraph 4.2.3.5c should reflect the anticipated maximum viewing distance. Using large fonts with broad stroke widths is recommended to improve readability. Selecting background color and contrasting foreground (text) color should ensure sufficient contrast for good readability.
- g. The usual designation of font size is given in points. The display of text fonts on screens is proportional to point size, but the actual size of displayed text is related to screen size and application software. The font point size only controls the actual size of printed output. It is recommended that screen text size be reviewed and adjusted in relation to the objective hardware system.

4.2.3.6 Scrolling of Data

- a. Display text information statically on the screen, rather than constantly scrolling it across the screen.

- b. If text is meant to be scanned by constant scrolling, columns with 35 characters or fewer per line are preferred.

4.2.4 GROUPING

4.2.4.1 Grouping for Data Comparison

If users must analyze sets of data to discern similarities, differences, trends, and relationships, structure the format of displays so the data are consistently grouped.

4.2.4.2 Important Data Placement

Where displayed data are used in some spatial or temporal order, consider grouping those data by sequence of use to preserve that order.

4.2.4.3 Primary Viewing Area

When data and terms are particularly important, require immediate user response, or are more frequently displayed, group them in the primary viewing area of the user.

4.2.4.4 Data Grouped Alphabetically or Chronologically

When there is no appropriate logic for grouping data by sequence, function, frequency, or importance; adopt some other principle, such as alphabetical or chronological grouping.

4.2.4.5 Arrangement of Data on Screen

Arrange and group data on application display screens to differentiate between instructions and data and to facilitate observation of similarities, differences, and trends for the most common uses.

4.2.4.6 Provide Cohesive Groupings

Provide cohesive groupings of screen elements by using blank space, surrounding lines, different intensity levels, etc.

4.2.5 MULTIPLE SCREENS

4.2.5.1 Paging Crowded Displays

When a display contains too much data for presentation in a single frame, partition the data into separately displayable pages. Refer to Section 5, Windowing.

4.2.5.2 Related Data on Same Page

When partitioning displays into multiple pages, take into account the type of data being partitioned, and display functionally related data items together on one page.

4.2.5.3 Multiple Pages Labeling

In a multipage display, label each page with a unique identifier that shows its relation to the other pages.

4.3 COLOR

As emerging command and control systems implement GUIs and high resolution color graphics displays, the use of color as an information discriminator has become crucial. Color can be a very effective discriminator, for example, by decluttering a display and improving task performance. Color can also induce the very clutter and performance degradation it attempts to reduce. For these reasons, color in a display must be used very carefully.

A designer must be sensitive to the many factors that affect how a person perceives and reacts to color as an information discriminator. An in-depth discussion on visual perception, color, and human performance is beyond the scope of this document; nevertheless, Table 4.1 provides definitions of key terms associated with the subject.

The designer should be aware of the following important guidelines:

Both brightness and type of lighting (e.g., incandescent versus fluorescent) can affect how colors are perceived. For example, bright ambient light desaturates display colors, leading to degraded color identification and discrimination. It may shift the eye's adaption, also

reducing the ability to discriminate color (Thorell and Smith 1990). In essence, identically colored objects can be perceived as being dissimilar under different lighting conditions.

- How the color of a foreground object is perceived is directly related to its background color.
- Visibility and readability are a direct result of the contrast between the foreground and the background.

Additional overarching guidelines for using color in computer display systems are:

- Use color sparingly as an information discriminator. Color rapidly loses meaning and, when overused, may impede rather than enhance human performance.
- Use colors consistently within a display and across a set of displays for an application.
- The meaning of color should be consistent with user expectation.

TABLE 4.1: Definitions of Key Terms for Color Usage

<u>TERM</u>	<u>DEFINITION</u>
Achromatic	Colorless; lights that have no definite hue.
Brightness	The perceptual (psychological) correlate of intensity that ranges from dark to bright.
Chromatic	Highly colored.
Discrimination	Degree to which a human visual system can sense differences in the physical characteristics of an image.
Hue	The psychological attribute of color sensation associated with the physical property of visible wavelengths.
Legibility	Ability to identify an alphanumeric character or symbol. A criterion of image quality.

Luminance	The amount of light reflected or emitted by a surface, measured in footLamberts.
Luminance Contrast	Ratio of the foreground brightness compared to the background brightness.
Monochromatic	Consisting of one color or hue.
Recognition	Ability to recognize or interpret the meaning or association of an image.
Saturation	The degree to which a hue differs from a gray of the same lightness.

- When using color to impart a specific meaning to the user, utilize an additional, redundant form of coding, such as shape. This ensures the correct meaning will be conveyed should the user have a color vision deficit or should color be unavailable on the screen.
- Standardize color coding for military applications. Whereas flexibility in color coding schemes may be desirable for a terminal dedicated to a single user, color coding should be standardized for military applications. Because of the variety of users on a tactical terminal, a terminal with a uniquely customized color coding scheme may be very difficult to interpret. The user should be allowed to select colors schemes for aspects of the application that do not involve coding or status. A default scheme should be easily available to restore the interface for subsequent users.
- A requirement for adjustable colors is created because of portable applications among hardware configurations. Each hardware system display has different color perceptual values and color names. Portable applications must be able to accommodate these differences. Status colors should be assigned during installation; the user should not be allowed to adjust these colors.

The following paragraphs provide more detailed guidelines for using color in DoD systems. Note that many of the guidelines contained in this section are most relevant to using color in text-based software, primarily because the majority of the past research in color usage was done with text applications. Whereas results of research with GUIs and graphical presentation of

information are emerging, more research is needed on using color in tactical graphics applications, especially in foreground/background combinations for colored map graphic displays.

The designer should also be aware that it is easier to define color combinations to avoid than to identify a single best way to utilize color. Color choice/combination tends to be a matter of personal preference. For example, high contrast between foreground text and background is crucial but can be accomplished with a number of color combinations. This ambiguity becomes all the greater when developing design guidance for the different command and control applications represented across military systems. The designer should utilize those guidelines most relevant to the particular application.

Color should be used when a basic monochromatic presentation of tactical information needs to be augmented for the user to gain a more effective understanding of the information being presented.

4.3.1 GENERAL

4.3.1.1 When to Use

Color should be used carefully (as a coding method, color can rapidly lose its effectiveness). Use color, when necessary, to:

- a. attach specific meaning to tactical information presented in the form of text or symbology
- b. direct the user's attention to the most important or time-critical information on the screen (i.e., information category headings, system and user errors, information requiring immediate attention, key data items, and window titles)
- c. enable a user to differentiate rapidly among several types of information, especially when the information is dispersed on the display
- d. increase the amount of information portrayed on a graphic display by adding color in addition to shape
- e. indicate changes in the status of graphical data.

4.3.1.2 Constraints on Use

The user with defective color vision will have difficulty discriminating among the colors. The following should be considered when including color in display screens:

- a. Add color coding only after displays have already been designed as effectively as possible in an achromatic format.
- b. Color only logically related information with similar hues. Consider spacing or highlighting instead of, or in addition to, color.
- c. When emphasizing tactical information by means of color, use a color for more important information that is brighter than adjacent color-coded information. Ensure the choice of colors is consistent with the user's expectations for the information being coded. See Section 4.3.2.
- d. Do not use color coding when it might confuse users with defective color vision or when the use of color reduces screen readability. If color must be used, consider the following:
 1. When the user must compare data, such as those contained in graphs based on color, avoid using green, yellow, and red as comparison colors for application information requiring important or frequent discriminations. If possible, use yellow and blue, or red and cyan.
 2. When green, yellow, and red must be used, provide other cues, such as brightness and saturation, to enhance discriminability.
 3. Do not code solely by color. Make color coding redundant with some other display feature, such as symbology or actual text content.
 4. Avoid requiring the user to discriminate between colors in small areas of the display. Small, color-coded areas are subject to loss and bleeding of colors. Use achromatic colors (i.e., black or white) if coding must be done in small areas.
 5. Use white to highlight data that require particular attention. Do not excessively use white as a highlighter, as it can create a glaring brightness that may interfere with screen legibility. When status changes are signaled by color, that color should not be used to highlight text. Status changes using color coding should be signaled by a ball or box next to the text.

6. Contrast should be high between the text or graphical object and its background to enhance screen readability. Generally, the color foreground should differ from its background by a minimum of 100 ΔE (CIE Yu'v') distances. Minimum luminance contrast ratios are required for specific tasks. For discrimination and legibility, acceptable ratios of foreground-to-background luminance contrast range from 6:1 to 10:1. Table 4.2 provides guidance for specific conditions. Using pure white or black as a background color is not recommended. Unsaturated hues provide the best background contrast.

TABLE 4.2: Recommended Luminance Contrast Ratios

<u>Condition</u>	<u>Ratio of Foreground to Background</u>
Bright Ambient Illumination	>7:1
To Attract Attention	>7:1
To Sharpen Edges	>7:1
Continuous Reading	3:1 to 5:1
Dark Ambient Illumination	3:1 to 5:1
Camouflage Images or Smooth Edges	<3:1

4.3.2 COLOR SELECTION

4.3.2.1 General

- a. When selecting colors for coding discrete categories of information displayed on a screen, ensure those colors are easily discriminatory in all expected operational environments.
- b. To aid in color discrimination, use colors that are as widely spaced along the visible color spectrum as possible. The following colors, listed by their wavelengths in millimicrons, are spaced widely enough for easy discrimination from one another.

<u>Color</u>	<u>Wavelength in Millimicrons</u>
Red	700

Orange	600
Yellow	570
Yellow-green	535
Green	500
Blue-green	493
Blue	470

- c. Use an unobtrusive color to display information used infrequently on a screen. Unobtrusive colors have shorter wavelengths.
- d. Use warm colors (colors with a longer wavelength, such as red or orange) to convey action or the requirement for a response. Use cool colors (colors with a shorter wave-length, such as blue or green) to convey status or background information.
- e. Each color should represent only one category of displayed data, as defined in Paragraph 4.3.2.6. A mismatch of color and color association slows recognition time and increases misidentification of words.

4.3.2.2 Consistency

- a. Color should be applied consistently from screen to screen and from application to application to ensure the user can make the proper interpretations. This is applicable both within and across military command and control systems. For example, do not use status colors as window borders unless status coding is intended.
- b. Color coding should be consistent with the interaction of the label's color and the color associations of the words in the label. For example, the word ENEMY, if color coded, should be red rather than green.
- c. Choose colors for coding based on conventional associations with particular colors. These should conform, if possible, to those specified in DoD documents, such as Army FM 101-5-1: Operational Terms and Symbols (1985b).

4.3.2.3 Number of Colors to Use

- a. Implement color coding conservatively, using relatively few colors to designate critical categories of displayed data and only where it will help user performance.
- b. Use no more than four colors at one time when using alphanumeric screens, with a maximum of seven total for all screens.
- c. Use four standard colors, with others reserved for occasional use. Only eight or nine highly saturated colors can be easily discriminated, so do not exceed this number. Extensive coloring creates a brighter-than-necessary display, with subsequent negative impact on user performance.

4.3.2.4 Pairing of Colors

Colors should be carefully paired on a screen to maximize human performance.

- a. Avoid simultaneous use of highly saturated, spectrally extreme color pairs on a display screen. Examples include such color pairs as red and blue, yellow and purple, or magenta and green. The effect is most significant with red and blue.
- b. To emphasize different tactical information in text and presentation graphics displays, use contrasting colors such as red and green or blue and yellow. However, in color choice, be consistent with the guidance provided in other paragraphs of this section.
- c. To convey similarity in tactical information in text and presentation graphics displays, use similar colors, such as orange and yellow or blue and violet.
- d. Contrast between foreground objects and background displays should be high. Black on light blue and blue on white are good general foreground and background combinations. However, blue should not be used for the foreground when fine detail resolution is required, such as in target recognition.
- e. Avoid using extensive coloring (e.g., many different colors) for the background, segments of the background, or particular regions surrounding individual characters or symbols.

4.3.2.5 Color Selection and Ambient Illumination

The level of ambient illumination directly affects the perceived brightness and hue of a color. Consider the following when designing a color display :

- a. Green provides good general visibility over a broad range of intermediate luminances.
- b. Use red under high ambient lighting but not in low lighting.
- c. Yellow provides good general visibility over a broad range of luminances.

4.3.2.6 Specific Color Meanings

Use the colors and associated meanings listed in Table 4.3 for designing military color coding. The military intelligence community uses the classification Bar Colors listed in Table 4.4.

TABLE 4.3: Color Codes and Associated Meanings

<u>Color</u>	<u>Meaning</u>
Green	Non-alert, forces or situation at acceptable condition, obstacles on map graphics
Blue	Friendly forces symbology
Red	Alert, forces or situation at critical condition, enemy symbology
Yellow	Forces or situation at marginal condition, caution, NBC areas on map graphics
Black	Friendly forces symbology

4.3.2.7 Using Blue

Blue as a background color is most effective for tasks performed at close distances.

- a. Because the eye is relatively insensitive to blue, blue lines or dots will be very difficult to resolve. Avoid using saturated blue for small lines or dots when the background is dark.
- b. Use saturated blue only for background features in a display, not for critical data.

TABLE 4.4: Classification Bar Color Codes and Associated Meanings

<u>Bar Color</u>	<u>Meaning</u>
Green	Unclassified
Blue	Confidential
Red	Secret
Orange	Top Secret
Yellow	Sensitive Compartmented Information

4.3.2.8 Use of Color Keys

If the screen designer must deviate from the color meanings provided in Tables 4.3 and 4.4 or use other colors, a key that explains the color meaning should be included on the display.

- a. This key should be readily visually accessible on the display without having to scroll or expand the screen or window
- b. This key should appear in the color being defined.

4.3.2.9 Large-Screen Display Periphery Colors

Avoid the use of red and green in the periphery of a large-scale display. Yellow and blue are good periphery colors.

4.3.3 TONAL COLOR CODING

4.3.3.1 Color Coding for Relative Values

When relative rather than absolute values of a variable are important, display gradual color changes of a single color as a tonal code to show the relative values of a single variable. Display a monochromatic shading rather than spectral codes (different colors).

4.3.3.2 Ordered Coding

Where different areas of a map are coded by texture patterns or tonal variation, order the assigned code values, such that the darkest and lightest shades correspond to the extreme values of the coded variable.

4.3.4 COLOR-CODED SYMBOLS

Symbols that are color coded should use the following guidelines:

4.3.4.1 Color-Coded Symbol Size

Color-coded symbols should subtend a minimum of 20 minutes of visual arc. The designer must determine the maximum viewing distance from the display, then calculate the minimum size of the object, using the formula:

$$\text{Visual Angle (Min.)} = \frac{(57.3) (60)L}{D}$$

where L = size of the object, and D = distance from the eye to the object. See Figure 4.9.

4.3.4.2 Color-Coded Symbol Brightness

Color-coded symbols should have a minimum luminance of one footLambert.

4.3.4.3 Refresh Rates

The minimum refresh rate for color-coded symbols should ensure a flicker-free display.

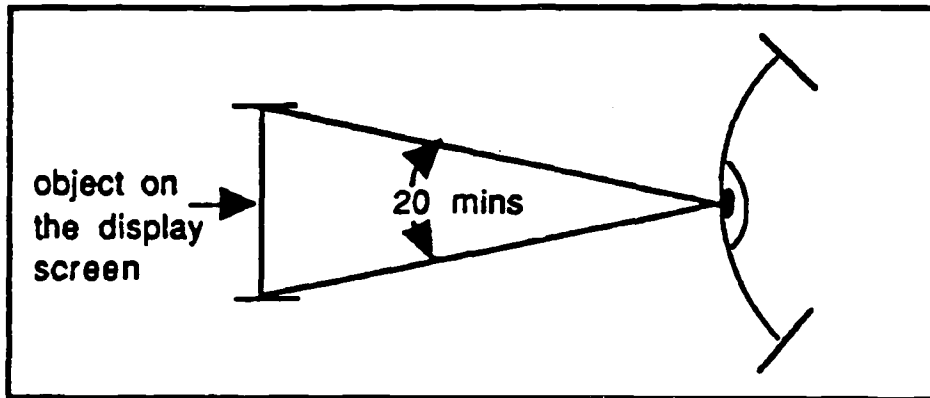


FIGURE 4.9: Visual Arc Subtended

4.3.5 MAP GRAPHICS AND COLOR

4.3.5.1 Functional Versus Decorative Color Coding

On map graphic displays, use color coding that provides a specific meaning to the user, rather than colors that are decorative only. These meanings should be in accordance with U.S. Army FM 21-26, Map Reading and Land Navigation (1987) (e.g., green for vegetation, brown for topographic relief, etc.).

4.3.5.2 Differences In Color Perceived Distance

The designer should be aware of how different colors focus at different distances relative to the user's retina as a result of wavelength. To the user, some colors will appear to be closer than others, especially the more saturated colors (see Paragraph 4.3.2.4a).

REFERENCE LIST

<u>Paragraph</u>	<u>Reference</u>
4.1	DoD (1992a)
4.2.1.1	Slominski and Young (1988) p. 2
4.2.1.2	Licktieg (1989) p.9; Brown et al. (1983) p. 1-10; Galitz (1984) p. 99 & 102; Tullis (1988) p. 382
4.2.1.3	Smith and Mosier (1986) para 2.5-7
4.2.1.4	Galitz (1984) p. 103; Shneiderman (1987) p. 327
4.2.2.1	Williams (1987b) Appendix A p. A-1; Smith and Mosier (1986) para 4.0-6; Brown et al. (1983) p. 1-11; Shneiderman (1987) p. 327; Smith and Mosier (1986) para 2.5-1; Licktieg (1989) p.10; Brown et al. (1983) p. 1-1 & 1-11, Tullis (1988) pp. 393 & 336; Hamel and Clark (1986) p. 26; Slominski and Young (1988) p. 2
4.2.2.2	Brown et al. (1983) p. 1-1
4.2.2.3	Williams (1987b) Appendix A p. A-2; MacGregor and Lee (1988) p. 10; Galitz (1984) p. 103; Shneiderman (1987) p. 336
4.2.3.1a	Smith and Mosier (1986) para 2.5-2
4.2.3.1b	Hamel and Clark (1986) p. 28; Slominski and Young (1988) p. 3-4
4.2.3.1c	Bowser (1991) p. 16; Tullis (1988) p. 385
4.2.3.1d	Galitz (1984) p. 103
4.2.3.1e	Shneiderman (1987) p. 327
4.2.3.2a	Nes (1986) p. 105
4.2.3.2b	Smith and Mosier (1986) para 2.5-10; Licktieg (1989) p. 10; Brown et al. (1983) p.1-5 & 1-12; Shneiderman (1987) p. 327
4.2.3.2c	Bowser (1991) p. 16; Smith and Mosier (1986) para 2.5-11; Galitz (1984) p. 102
4.2.3.2d	Tullis (1988) p. 382
4.2.3.2e	Galitz (1984) p. 102; Licktieg (1989) p. 9
4.2.3.2f	Brown et al. (1983) p. 1-4
4.2.3.3a	Williams et al. (1987b) Appendix A p. A-1
4.2.3.3b	Smith and Mosier (1986) para 2.5-3
4.2.3.3c	Nes (1986) p. 103; Shneiderman (1984) p. 104; Brown et al. (1983) p. 1-6
4.2.3.3e	Nes (1986) p. 101; Brown et al. (1983) p. 1-13; Shneiderman (1987) p.105; Tullis (1988) p. 398-399; Grabinger and Amedeo (1988) p. 198
4.2.3.3d	Shneiderman (1987) p. 327

REFERENCE LIST (Cont.)

<u>Paragraph</u>	<u>Reference</u>
4.2.3.3f	Tullis (1988) p. 395
4.2.3.3g	Galitz (1984) p. 102
4.2.3.3h	Shneiderman (1987) p. 327; Brown (1989) p. 28-29
4.2.3.4a	DoD (1985) p. 3-3
4.2.3.4b	Tullis (1988) p. 399; Galitz (1984) p. 104; DoD (1985) p. 3-3
4.2.3.5a	Shneiderman (1987) p. 104; Nes (1986) p. 112; Tullis (1988) p. 397; Brown et al. (1983) p. 1-9
4.2.3.5b	Galitz (1984) p. 184
4.2.3.5c	Galitz (1984) p. 184
4.2.3.5d	Shneiderman (1987) p. 184
4.2.3.5e	Bowser (1991) p. 16
4.2.3.5f	Bowser (1991) p. 16
4.2.3.6a	DoD (1985) p. 3-2
4.2.3.6b	DoD (1985) p. 3-3
4.2.4.1	Smith and Mosier (1986) paras 2.5-13 and 2.5-15; Tullis (1988) p. 387; Shneiderman (1987) p. 336
4.2.4.2	Smith and Mosier (1986) para 2.5-14
4.2.4.3	Smith and Mosier (1986) paras 2.5-16 and 2.5-17
4.2.4.4	Smith and Mosier (1986) para 2.5-18
4.2.4.5	Brown et al. (1983) p. 1-11
4.2.4.6	Galitz (1984) p. 102
4.2.5.1	Smith and Mosier (1986) para 2.5-4
4.2.5.2	Smith and Mosier (1986) para 2.5-5; Galitz (1984) p. 103; Shneiderman (1987) p. 327
4.2.5.3	Smith and Mosier (1986) para 2.5-6; Shneiderman (1987) p. 327; Brown et al. (1983) p. 1-5
4.3.1	Lickteig (1989) p.10; Nes (1986) paras 4.2.3 and 4.2.4; Galitz (1984) p. 122; Brown et al. (1983) para 7.2; Lickteig (1989) p. 10; Shneiderman (1987) p. 341; Bailey (1982) p. 421; Lewis and Fallesen (1989) p. 20
4.3.1.2a	Smith and Mosier (1986) para 2.6-29
4.3.1.2b	Shneiderman (1987) p. 72 and 337; Tullis (1988) p. 390
4.3.1.2c	Lewis and Fallesen (1989) p. 23; Nes (1986) para 4.1.1; DoD (1989) para 9-1.3.3; Galitz (1984) p. 126 and 127; Slominski p. 4; Matthews (1987); Sidorsky p. 6.3-15q

REFERENCE LIST (Cont.)

<u>Paragraph</u>	<u>Reference</u>
4.3.1.2d	Galitz (1984) p. 126-127
4.3.1.2e	Sidorsky p. 2.3.6 q-6,7; Galitz (1984) p. 121; Bailey (1982) p. 43; Lewis and Fallesen (1989) p.25
4.3.1.2f	Smith and Mosier (1986) para 2.6-30; Brown para 7.4; DoD (1989a) para 5.4.1.4.5.5; Bailey (1982) p. 63, Lewis and Fallesen (1989) p. 20; Lickteig (1989) p. 10
4.3.1.2g	Brown et al.(1983) para 7.7.6
4.3.1.2h	Bowser (1991) p.18; Lewis and Fallesen (1989) p. 21; HFS (1988)
4.3.2.1a	Smith and Mosier (1986) para 2.6-27
4.3.2.1b	Galitz (1984) p. 125; Lewis and Fallesen (1989) p. 22
4.3.2.1c	Galitz (1984) p. 123
4.3.2.1d	Lewis p. 22
4.3.2.1e	Smith and Mosier (1986) para 2.6-26, 31; Sidorsky p. 6.3-15 o-1; Brown para 7.6.1; Nes (1986) para 4.2.2; Galitz (1984) p. 122; DoD (1989) para 5.15.3.3.7; Lickteig (1989) p. 10; Shneiderman (1987) p. 339
4.3.2.2a	Bowser (1991) p. 18; Brown (1983) para 7.6.2; Galitz (1984) p. 125; Shneiderman (1987) p. 340, Lewis and Fallesen (1989) p. 21; Bailey (1982) p. 263
4.3.2.2b	Bailey (1982) p. 246
4.3.2.2c	Smith and Mosier (1986) para 2.6-32; Sidorsky p. 6.3-15 o-2 and 2.3.6 q-3, 4; Brown (1983) para 7.7.1; Galitz (1984) p. 125; DoD (1989) para 5.2.2.1.18; Shneiderman (1987) p. 340; Hamel p. 5; Bailey (1982) p. 246, Lickteig (1989) p. 10
4.3.2.3d	Smith and Mosier (1986) para 2.6-28; Brown (1989) para 7.1; Galitz (1984) p. 127; Lickteig (1989) p. 10; Smith and Mosier (1986) para 2.6-28; Chao (1987) p. 361; Lewis and Fallesen (1989) p. 20
4.3.2.3e	Galitz (1984) p. 127; Nes (1986) para 4.2.2 and 4.2.5; Slominski p. 4; Shneiderman (1987); p. 338
4.3.2.3f	Sidorsky p. 6.3-15 o-3; Galitz (1984) p. 125;Lickteig (1989) p. 10; Bailey (1982) p. 421; Shneiderman (1987) p. 71
4.3.2.4a	Lewis and Fallesen (1989) p. 22
4.3.2.4b	Galitz (1984) p. 126

REFERENCE LIST (Cont.)

<u>Paragraph</u>	<u>Reference</u>
4.3.2.4c	Galitz (1984) p. 126
4.3.2.4d	Lewis and Fallesen (1989) p. 22; Shneiderman (1987) p. 341; Snyder (1988) p. 465; Sidorsky para 2.3.6; Thorrell p. 2 and 3
4.3.2.4e	Shneiderman (1987) p. 341; Snyder (1988) p. 465; Matthews (1987) p. 23
4.3.2.4f	IBM (1984) p. 19
4.3.2.5a	Galitz (1984) p. 127
4.3.2.5b	Galitz (1984) p. 127
4.3.2.5c	Galitz (1984) p. 127
4.3.2.6	Sidorsky p. 2.3.6 q-3 and 4; DoD (1989) p. 256; U.S. Department of the Army (1985) p. 2.2
4.3.2.7	Lewis and Fallesen (1989) p. 23
4.3.2.7a	Snyder (1988) p. 465
4.3.2.7b	Smith and Mosier (1986) para 2.6-34; Brown (1983) para 7.7.5; Galitz (1984) p. 127
4.3.2.8	Sidorsky p. 6.3-15 o-4; Brown para (1983) 7.6.1; Nes (1986) para 4.2.4; Galitz (1984) p. 123; Lickteig (1989) p. 10; U.S. Department of the Army (1985) p. 2-2
4.3.2.9	Lewis and Fallesen (1989) p. 22
4.3.3.1	Smith and Mosier (1986) para 2.6-25
4.3.3.2	Smith and Mosier (1986) para 2.4.8- 7
4.3.3.3	Smith and Mosier (1986) para 2.4.8-8
4.3.4.1	Durrett (1987) p. 186; Van Cott and Kinkade (1984) p. 47
4.3.4.2	Breen et al. (1987) p. 207
4.3.4.3	Breen et al. (1987) p. 209
4.3.5.1	Olson (1987) p. 207; U.S. Department of the Army (1987)
4.3.5.2	Olson (1987) p. 209

5.0 WINDOWS

A window provides the visual means by which the user can interact with an application program. Results of the command or data input by keyboard, mouse, or other device will be displayed in the window. A window display screen is analogous to a window in a wall that allows one to see into a room; the window display screen allows the user to see into a software program. Typically, a window is rectangular and can cover part or all of a display screen. In addition, multiple windows can be open at one time on a display. Figure 5.1 illustrates a typical window. The arrows and labels identify key parts of the window. This section provides general guidelines for windows. Refer to the specific GUI style guides for a detailed explanation of attributes and the terms used to describe the actions, warnings, and information presented to the user. The user controls window operation using a number of basic functions. By

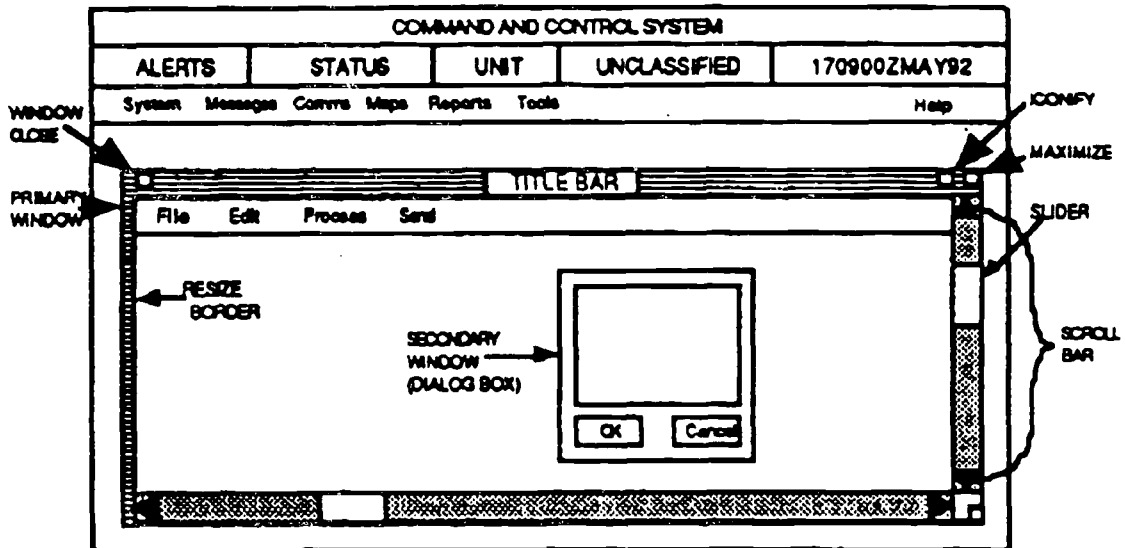


FIGURE 5.1: Example of a Typical Windowing Screen

opening and closing a window, a task or application is started, stopped, or removed from the screen. Scrolling allows the user to view the information within a window, including that which is outside the normal boundaries of the window. Windows can be stacked on top of each other like paper on a desk. Good designs should provide the user the capability to reshuffle and

move windows relative to each other (BRING-TO-FRONT AND PUSH-TO-BACK commands).

The two basic approaches to simultaneous window presentation are tiling and overlapping. In the tiling approach, multiple windows do not overlap but lie on the same plane. Their borders are flush, and the primary control operations that can be applied to each window are opening and closing, designation, scrolling, moving, and sizing. Figure 5.2 illustrates this type of window design. Using the overlapping method, windows are presented on multiple planes and

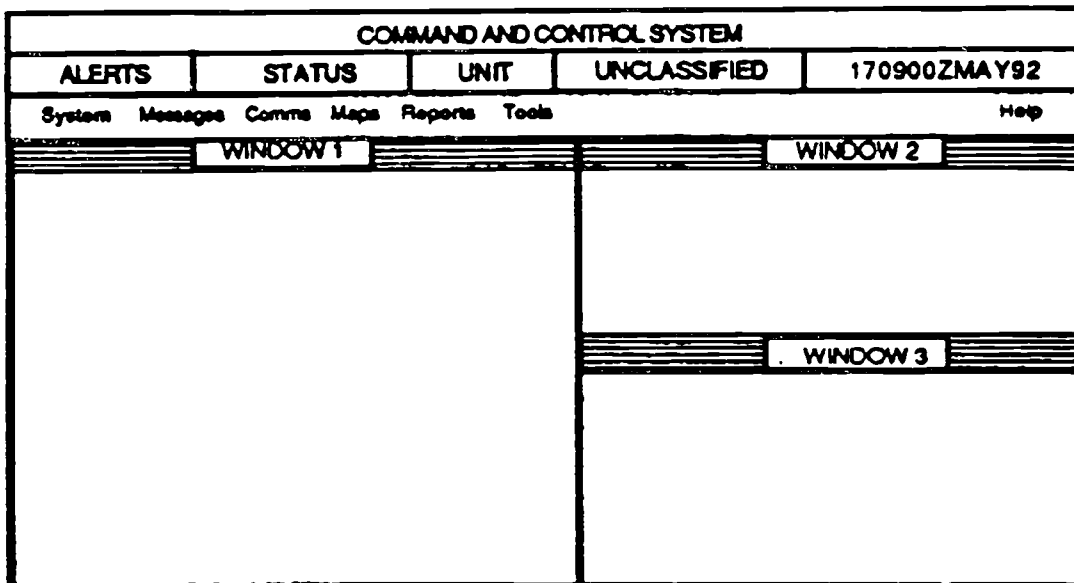


FIGURE 5.2: Example of the Tiling Approach

appear to be three-dimensional. Windows can overlap or even obscure each other, like pieces of paper on a desk top. The user can apply all of the control functions previously discussed to overlapping windows. Figure 5.3 illustrates this type of window design.

One difficulty in developing design guidelines for windowing is the lack of information, primarily because the design phenomenon is so recent. The depth and breadth of research on how windows impact user performance are not as great as they are for other design areas (Billingsley 1988). In general, the following generic guidelines can be applied to the design of windowing:

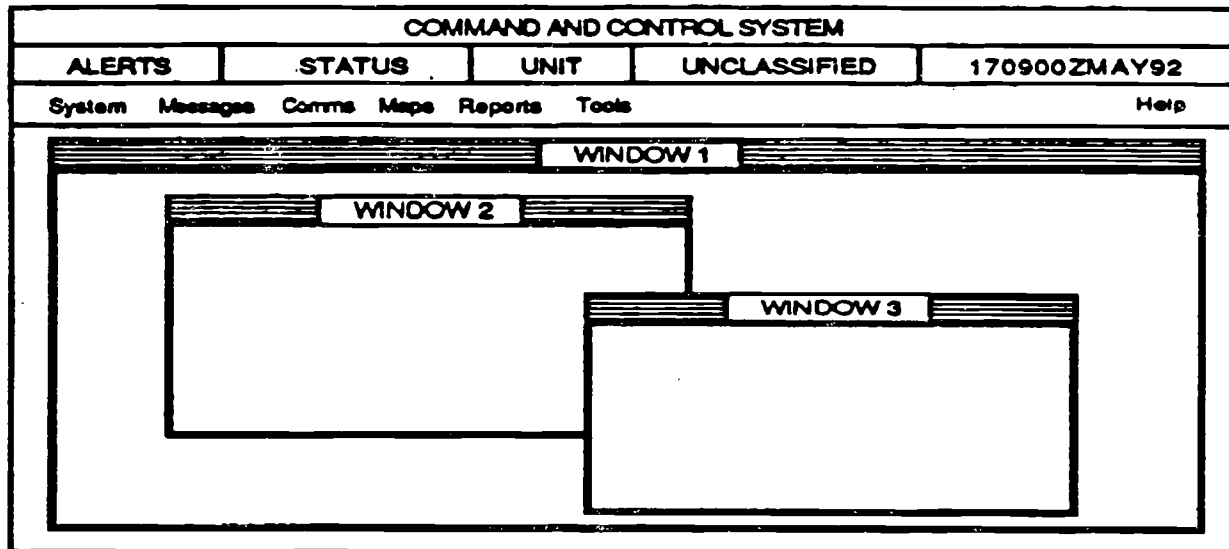


FIGURE 5.3: Example of the Overlapping Approach

- Be consistent in how the windows look and "act."
- Ensure the display device has the resolution and size to support properly a windows approach to information presentation.
- Ensure the central processor has the power, in terms of memory and speed, to utilize effectively a windows approach. Without proper hardware, the speed of information presentation will be significantly degraded by slow system-response time.
- Use windows if the user needs to perform multiple tasks or see different sets of data concurrently.
- The designer must recognize the limitations that the hardware imposes on the usefulness of windowing software.
- Each open window requires system resources in terms of memory and processing speed. A limit on the maximum number of windows that can be effectively opened for each system should be determined through experimentation.

5.1 WINDOW BASICS

5.1.1 BASIC WINDOW APPEARANCE

Although within application areas such as intelligence the basic Compartmented Mode Workstation (CMW) window components are the same, Motif and Open Look windows differ significantly. For example, the bottom line of the base window is called the message area in Motif implementations and the footer in Open Look implementations (see Section 2). These differences are not just semantic; they also vary in the functions and features that can be implemented.

The classification bar displayed as the top line of the basic window and the optional input information label displayed at the bottom of the screen are value-added features supported by the intelligence community CMW operating system rather than by the GUI style selected. However, from the CMW application designer's viewpoint, the classification bar and input information label are displayed in the same manner as the other window controls (e.g., the title bar). A detailed description of the fields that make up these security bars can be found in Appendix A. Until the CMW becomes an integral component of the DoD system architecture, each DoD organization should adhere to its own security standards.

5.1.1.1 Title Bar

The title bar in Motif displays the window title and three control buttons: the window menu button, the minimize button, and the maximize button. The title bar in Open Look displays the window title and the window menu button. The title should clearly identify the application to the user. Some general considerations that apply to creating titles are as follows:

- a. The title should contain the name of the application, followed by a colon, followed by the name of the currently displayed file or menu (e.g., Editor:Myfile.txt).
- b. The title should be centered.
- c. The title should be distinguished by a visual attribute (e.g., boldface type).
- d. Secondary windows should identify the application and the window's function (e.g., "Trusted Path: Authorization Menu ISSO").

- e. A window's title may display the version number of the application, but should not display any messages.

5.1.1.2 The Window Menu Button

The window menu button is located in the upper left-hand corner of the title bar (see Figure 5.1). This button provides a standard location for window management functions (e.g., close, move, and window resizing functions). A more detailed explanation of the functions and features supported by the window menu button can be found in the relevant GUI style guides.

5.1.1.3 Reducing the Window to an Icon

A base window that a user wishes to keep open but is not using actively or an active window not requiring user interaction for extended periods can be reduced to icons. When a window is reduced to an icon, the window is removed from the screen, and the application controlling the window is represented as an icon. Application processing then continues in the background, as if the window were still displayed on the screen. In OSF/Motif applications, windows can be reduced to icons by selecting the minimize button from the title bar, selecting the minimize function from the window menu bar, or depressing the minimize accelerator keys with the window focus appropriately selected. In Open Look applications, windows can be reduced to icons by selecting the Close function in the base window menu. Icons should be opened by positioning the pointer over the icon and double-clicking the Select button on the mouse.

5.1.1.4 Expanding a Window to its Full Size

Expanding a window to its full size (maximizing) increases the size of the window to the maximum specified by the application. In OSF/Motif applications, windows can be maximized by either selecting the maximize button from the title bar, selecting the maximize function from the window menu button, or depressing the maximize accelerator keys with the window focus appropriately selected. In Open Look applications, windows can be maximized through the base window menu. Windows can also be expanded to full size by dragging the resize borders or resize corners.

5.1.2 DRAGGING THE WINDOW

"Dragging" refers to the user's ability to reposition windows on the screen. Dragging a window moves it to a different position on the computer screen. A Motif window can be dragged using the mouse by positioning the mouse pointer over the title area of the title bar, pressing the Select button on the mouse, moving the mouse pointer to the desired location, and releasing the button. An Open Look window can be moved by dragging the window background, where the background is any part of a window that is not covered by a control, pushpin, or scroll bar. (A window can also be moved using the appropriate keyboard function key, or using the move option from the window menu button.) As the window is dragged (or moved), a "ghost" outline of the window should move with the pointer. The window should move the position of the outline when the mouse button is released.

5.1.3 SCROLL BARS

The scroll bar is a special type of control that makes it easy for the user to view or page through objects (such as documents, drawings, and spreadsheets that are too long or wide to be displayed in the application area or pane) and pan graphic map displays in the north/south and east/west directions. Scroll bars give users the capability to navigate through documents without paging one window at a time.

5.1.4 APPLICATION AREA

The application area or pane is the part of the window where applications display and collect data and where users perform most application tasks. For example, if a user is working with a text editor, the application area could contain the document to be edited.

5.1.5 MESSAGE AREA

The message area (or footer) is reserved for non-critical application messages that should not suspend processing. The left side of the message area should be used for short-term messages such as "Incorrect format - field requires numeric data. Please reenter." The right side of the message area should be used for medium-term messages such as "Page 4 of 29."

5.1.6 RESIZING THE WINDOW

The application suggests the initial size of its window to the window manager. Because work and preferences vary, users should generally be able to alter the size of windows. Resizing windows is performed through "hooking" the edge or corner and dragging the cursor to reduce or increase the window size, or by using buttons typically located in the upper right corner of a window.

Resizing a window normally increases or decreases the size of the window frame, not the scale of the data within the window. For example, if a window containing a text document is enlarged, more lines of data may be seen, but the text itself does not enlarge.

In a window that has multiple panes, it is up to the application designer to define exactly how the space within the window will be divided between the panes after the user resizes the window. Suppose, for example, a window contains overlapping panes as in Figure 5.3. When the user makes the window taller, the additional height can be distributed entirely in the bottom pane, entirely in the top pane, equally in the two panes, or in proportion to the relative sizes of the two panes.

Some guidelines to follow when resizing windows are:

- a. The minimum height of a window should allow enough room for at least the classification bar, title bar, and menu bar (control area).
- b. The application should be logically designed to accommodate the resizing function. Pertinent information should be contained in the upper left-hand corner of the window.
- c. When a user resizes a window, only the size of the window's borders should change, not the size or relative position of the data or the controls within the borders. An exception might occur in imagery manipulation where the user may require the image to rescale (magnify) with the window frame.
- d. Resizable windows should be easily distinguishable from those, such as the system window, that cannot be resized. OSF/Motif windows normally have a wide frame border, made up of corner handles and edge handles; Open Look windows are drawn with resize corners. Users can drag the resize border or resize corner when they want to change the window's size.

5.1.7 WINDOW CONTROLS

Controls and their labels represent application functions in windows and dialog boxes. Controls should mimic the physical items they represent (e.g., switches or buttons) by providing feedback before, during, and after selection by a user. For example, a button that the user has chosen should appear to be pushed in.

Window controls are generally activated using the SELECT button on the mouse. However, users who interact with the application using only the keyboard should have equivalent functionality. Arrow keys should allow the user to move between controls, and pressing the Return/Enter key should invoke the indicated control. In addition, mnemonics should be provided for each control.

5.1.8 WINDOW COLORS/PATTERNS/AUDIO SIGNALS

The proper use of color, background patterns, and sound may significantly aid the user. This section provides recommendations for using these features.

- a. Color should always be redundant with some other visual attribute; color should not be provided as the only means of visual distinction.
- b. On both color and monochrome displays, background patterns can be used to highlight, group, or clarify relationships and to add extra meaning.
- c. For quick and accurate interpretation, colors should be used sparingly and should match user expectations.
- d. Colors should not be "hard coded" into applications. Users should have the option to select their own color schemes (see also Section 4.3).
- e. Some colors have strongly associated meanings. For example, a user may assume a red control button has critical or irreversible consequences. Red should thus be avoided for non-critical buttons, as it may inhibit the user from exploring them. Some common color meanings are as follows:

- Red Stop, alarms, errors, danger, critical consequences
- Yellow Warning, caution, approaching critical
- Green Normal, safe, within normal range, proceed
- Blue Cold, water, non-critical items
- Gray Inactive, unavailable options or actions

- f. Both color and sound should be used for messages that require user acknowledgement. Critical messages should be displayed in red, and the audio alarm should continue until the user responds. Non-critical messages (e.g., "Printer error. Please check printer and retry or cancel") should be displayed in yellow and should be accompanied by a short audio alert.
- g. Spectral extremes (e.g., red and green) should not be used together. Colors at considerably different wavelengths appear to vibrate when placed together.
- h. When data are color-coded, a legend (e.g., "Orange = Required Field") should be provided at the bottom of the window. Color codes should be limited to four per window and no more than seven per application.
- i. The same color scheme (window background, foreground, etc.) should be used for all windows of an application. Repeated use of the same color for similar user interface components or data types allows elements to be associated quickly.
- j. White text on a black background produces halation, or the spreading of light, making the text less readable. Text should only be displayed in multiple colors if the other colors provide additional meaning. Due to the inherent focusing problems with blue, it should never be used as a text color or for any critical item.
- k. The workspace or computer screen background should be a neutral color (preferably gray).
- l. The application window background should contrast sufficiently to stand out in the workspace foreground. At the same time, it should provide a neutral background for the application data to ensure readability. Muted pastels are recommended.
- m. In general, the larger the object, the less saturated or deep its color

should be to avoid eye fatigue.

- n. CMW Classification Bar colors are listed below. Restrict background colors that match these colors.

- Green	Unclassified
- Blue	Confidential
- Red	Secret
- Orange	Top Secret
- Yellow	Sensitive Compartmented Information

5.2 WINDOW DESIGN

5.2.1 GENERAL GUIDANCE

5.2.1.1 Hardware Limitations on the Use of Windowing

Avoid windowing when the hardware has the following limitations:

- a. small screen size, resulting in frequent manipulation of the screen by the user
- b. slow processing speed, resulting in slow operation of tactical functions performed by the computer
- c. low screen resolution, resulting in less effective visual coding, especially for tactical map graphics symbols and icons.

5.2.1.2 Flexibility of Window Specification

A key to the effective design of a windowing user-computer interface is the flexibility the user has in customizing window content and format. A balance must be achieved between user-specified windows and preformatted windows.

- a. When the need to view several different types of data jointly cannot be determined in advance, allow a user to specify and select separate data windows that will share a single display frame.
- b. Where the information required for decision making may vary according to the tactical situation, allow the user to specify what information to include in a display.

- c. When content of particular tactical displays can be determined during interface design, provide the user with preformatted windows, such as standard message texts for data entry and display.
- d. Allow the user to display several of these windows concurrently, according to the tactical need.

5.2.1.3 Temporary Window Overlays

Temporary window overlays are especially effective for providing a menu of alternatives for field entry in preformatted tactical messages and database queries.

- a. When it is necessary to temporarily add requested data or other features to a current display, provide window overlays for that purpose.
- b. Ensure a temporary window overlay does not completely cover the active window, thereby obscuring critical control information and command entry widgets, soft keys, or other activation points.
- c. When a window overlay temporarily obscures other displayed data, ensure the obscured data are not permanently erased but will reappear when the overlay is later removed.

5.2.1.4 Number of Allowable Open Windows

To ensure that system response time is not compromised, design into the command and control system a defined upper limit on the number of windows allowed to be open at one time.

5.2.1.5 Window Physical Design

- a. Avoid visual clutter in designing windowing systems.
- b. For tiled window systems, minimize the clutter at the edges caused by scroll bars, etc. Figure 5.4 illustrates a cluttered window design for tiled windows. Figure 5.2 illustrates an uncluttered display.
- c. For overlapping window systems with multiple windows, keep the background pattern neutral, rather than using complex patterns. See Figure 5.5 for an example of a cluttered display. Figure 5.3 illustrates an uncluttered display.

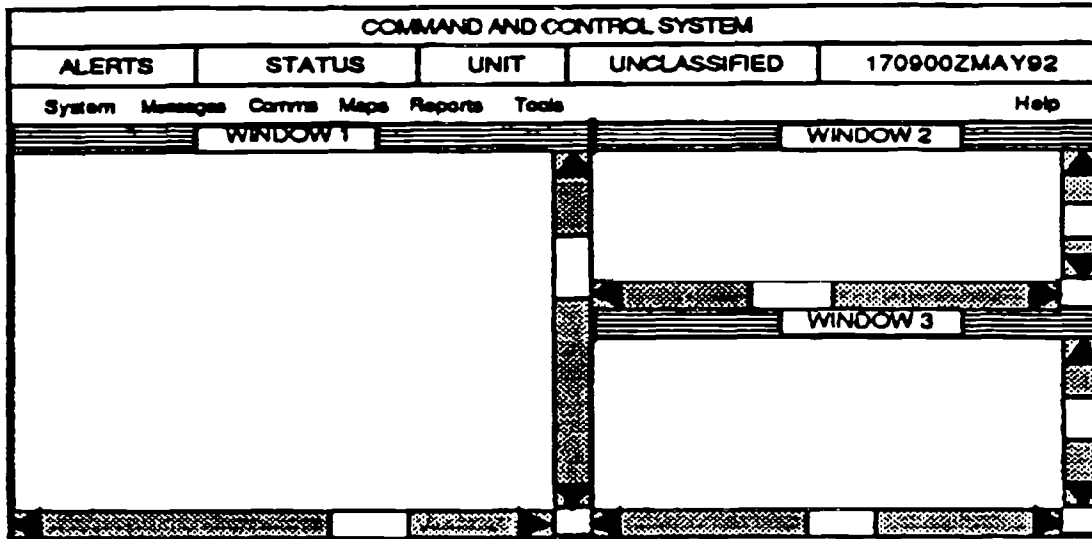


FIGURE 5.4: Example of Cluttered Window Design for Tiled Windows

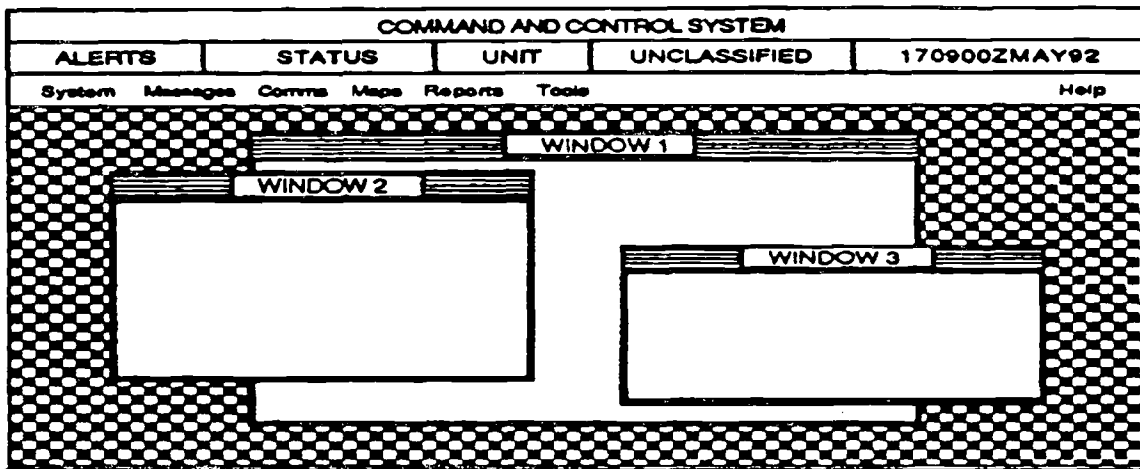


FIGURE 5.5: Example of Cluttered Window Design for Overlapping Windows Induced by a Complex Background Pattern

- d. When a display window must be used for scanning data that exceed more than one line, ensure the window can display more than one line of data.
- e. When the system provides an area within a window for command entry, messages, or prompts, place this area at the bottom of the window display.
- f. Dialog boxes should be designed to look and function consistently for all applications and systems. To achieve this, follow these recommendations (see Figure 5.6):

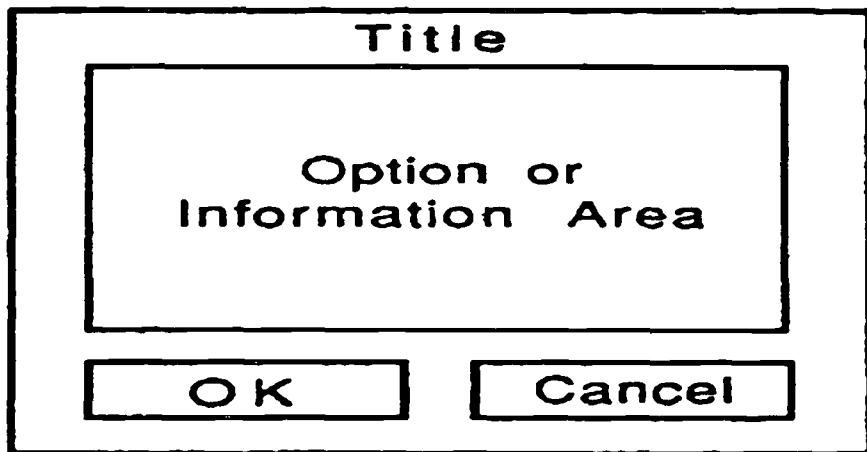


FIGURE 5.6: Example of How a Dialog Box Should Be Designed

1. Control buttons used to input a command from a dialog box should be located at the bottom of the window, making it consistent with the user's natural task flow.
2. The button used to input the selected or default command (usually an OK) should be located on the left side of the box. The CANCEL button should be located on the right side. Any additional control buttons should be located between the OK and CANCEL buttons.

5.2.2 WINDOW CONTROL

Control refers to how the user manipulates the window, not how the application operates within the window. Guidelines for the design of window control fall into five basic topics: general guidelines, opening and closing, moving, sizing, and scrolling.

5.2.2.1 General

- a. When a user may perform control actions (such as command entry) while working within a window, ensure those control actions will be consistent from one window to another.
- b. Ensure the means provided to the user for controlling the size, location, and characteristics of window overlays operate consistently from one display to another for each type of overlay.
- c. Provide an easy means, such as iconization or closing, for the user to suppress the display of window overlays.
- d. Provide a separate menu bar for each application window, where different applications are operating concurrently in open windows (e.g., multitasking). See the example in Figure 5.7.

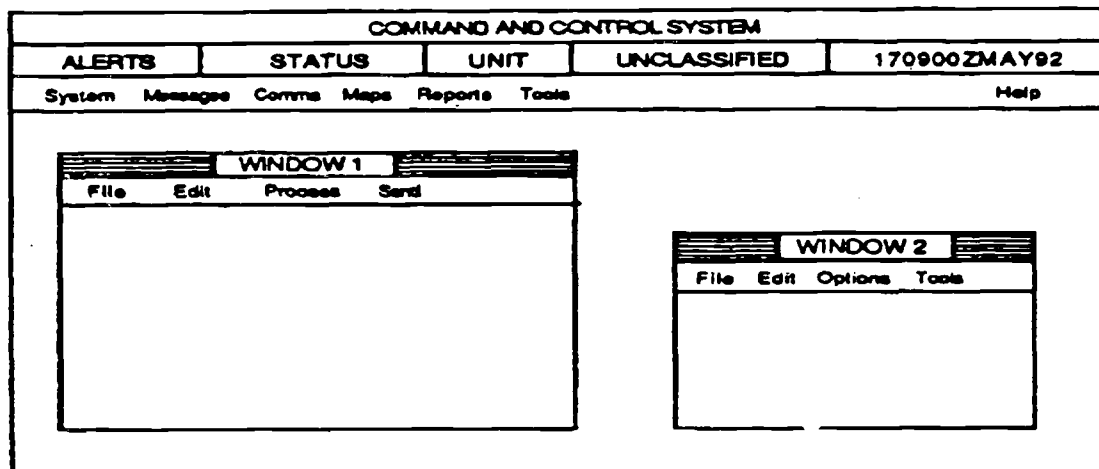


FIGURE 5.7: Example of Applications Menu With Menu Bars

5.2.2.2 Opening and Closing Windows

Windows are opened and closed by icons, menu selections, or a close-button widget (a small, push-button control object usually located in the upper left corner of a window). When designing the opening and closing operations,

consider the following guidelines:

- a. The software should provide an animated depiction of the opening and closing of a window by portraying the window shrinking to an icon and vice versa. This helps the user relate the window, icon, and action. See Figure 5.8.

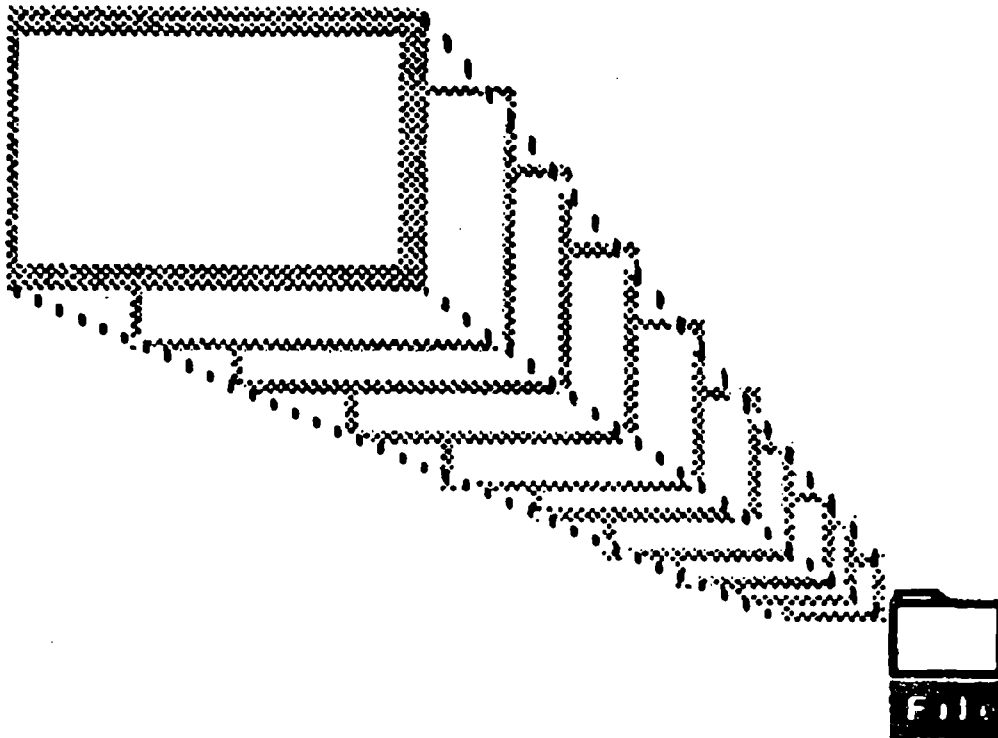
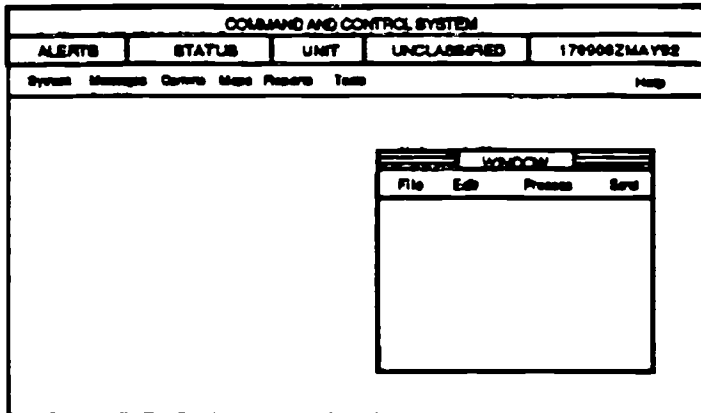


FIGURE 5.8: Example of Window Animation

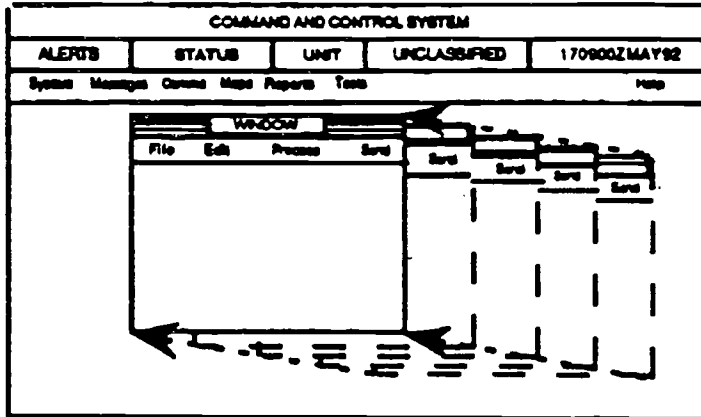
- b. When a main applications window is closed by the user, all associated subordinate windows and dialog boxes should also close.

5.2.2.3 Moving Windows

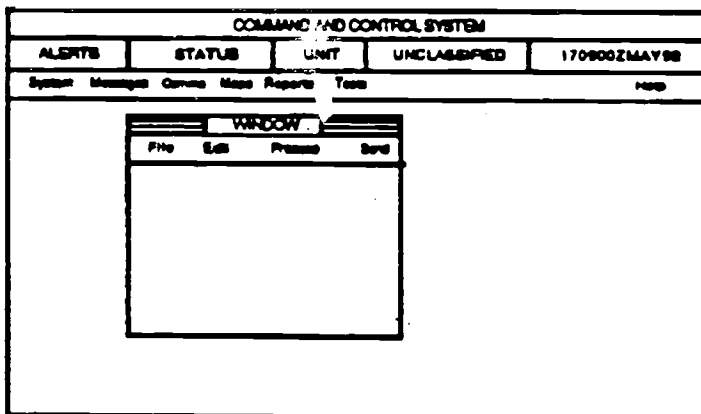
- a. Provide either full movement of the window (see Figure 5.9) or move an outline, leaving the window visible on the screen.



Original Location



Movement Indication



Final Location

FIGURE 5.9: Example of a Screen Move

- b. When the user must select a specific move function to relocate a window on a screen, ensure the cursor indicates this by a change in shape. Figure 5.10 illustrates one type of cursor change.

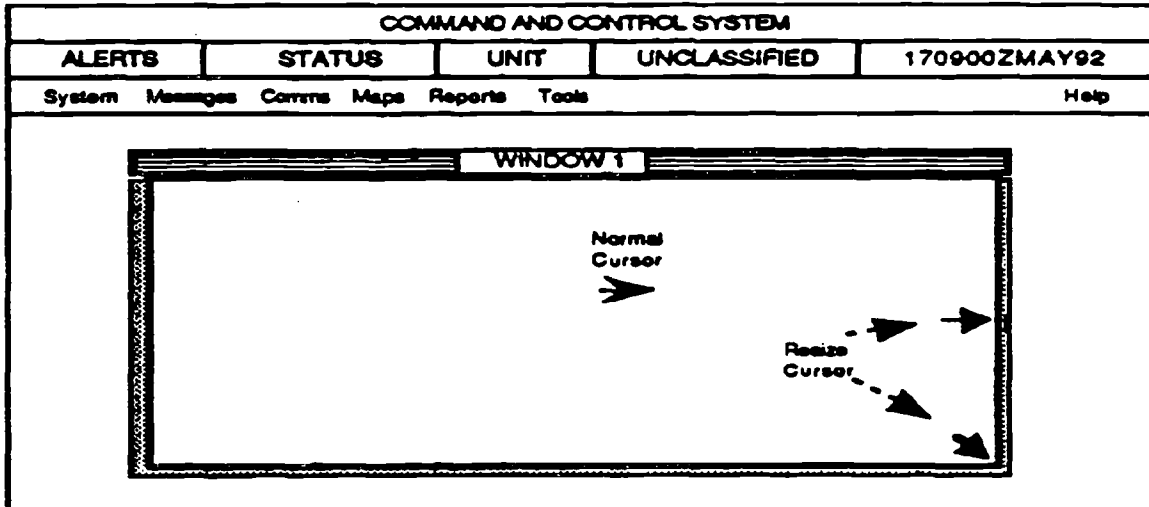


FIGURE 5.10: Example of a Cursor Changing Shape

5.2.2.4 Resizing Windows

- a. Provide system protection against obscuring critical control information during window manipulation, especially during user maximization of the window. This means system protection for both the data being retrieved through dialog boxes and system-level control information, such as alert indications. See Figures 5.11 and 5.12.
- b. When a window is resized, ensure the window contents remain visible during the resizing to provide a visual indication of the effect on the window contents (e.g., visibility and integrity of the image), rather than providing just an outline. Keeping the contents visible will reduce the number of steps required by the user (e.g., resize, view, etc.).
- c. Resizing of tiled windows by a user is not recommended. If resizing is absolutely required for a tiled window system, ensure the system automatically resizes all other open windows when one is resized by the user.

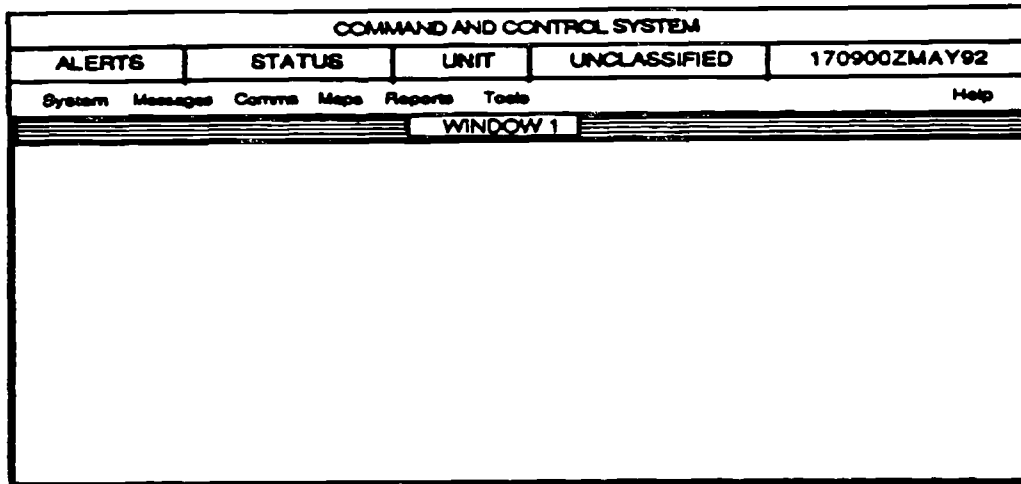


FIGURE 5.11: Maximum Size of Window

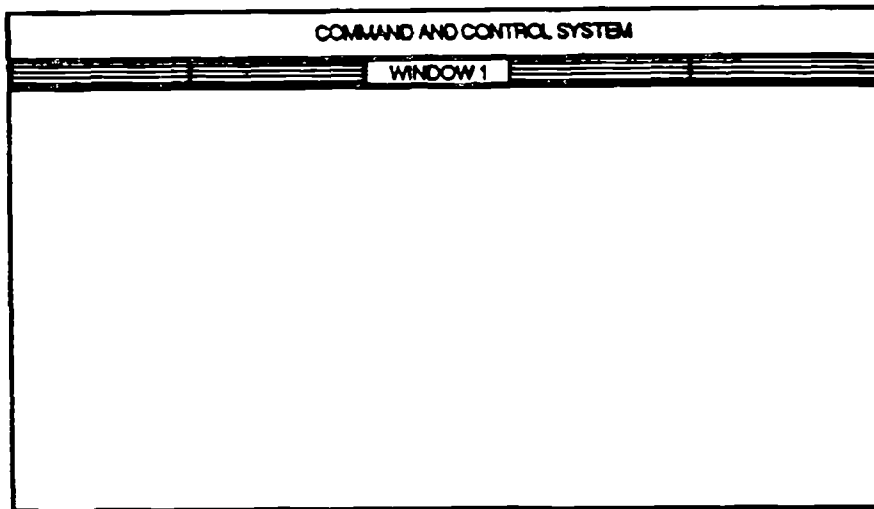
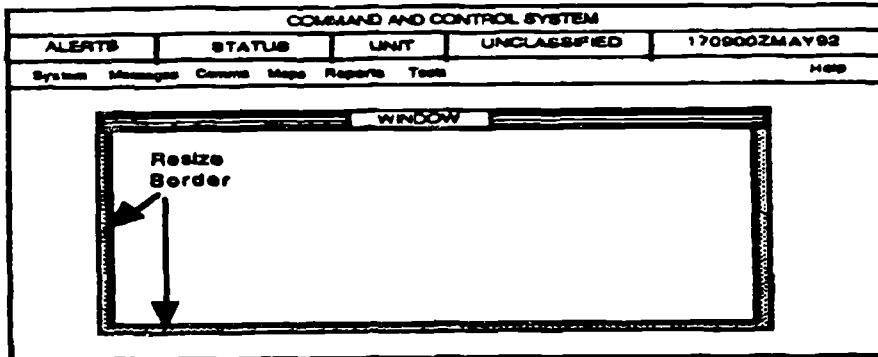
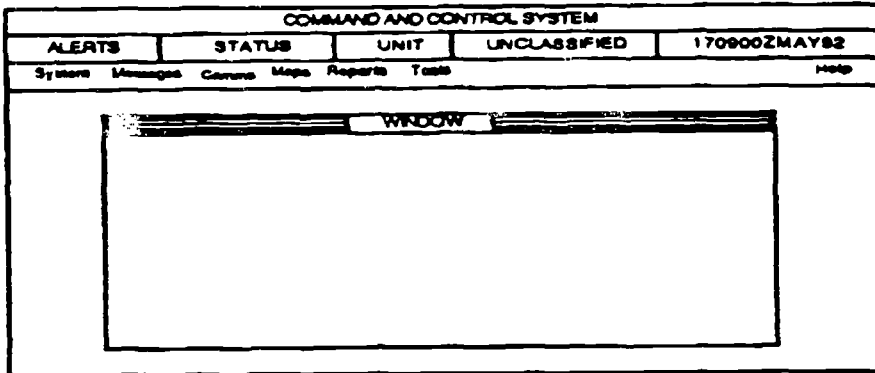


FIGURE 5.12: Window Size Too Large, Covering Critical Information

- d. Most windows have a resize border (Figure 5.13) located at the peripheral edge. If a window cannot be resized, the resize border should be removed to provide a positive indication to the user that the window size is static.



Window with a Resize Border



Window with Resize Border Removed

FIGURE 5.13: Resize Border Removal

5.2.2.5 Scrolling Windows

Scrolling a window can be performed two ways: 1) move the window over the data, where upward movement of the scroll bar causes data to appear to move down; 2) move the data past the window, where upward movement of the scroll bar causes data to appear to move up.

- a. For scrolling, the system should move the window over the data, as this is consistent with the general convention in industry and the OSF/Motif Style Guide.

- b. The distance the slider moves on a scroll bar should be proportional to the distance traveled through the file in a window to assist the user in determining current location relative to the total file.
- c. Design window displays to preclude excessive scrolling. If possible, use a single screen for the full display, unless it causes reading difficulty due to reduction of screen character size. If scrolling is not necessary the scroll bars should be removed.

5.2.3 DESIGNATION

Designation is the process of selecting and indicating with visual cues which window the user can use. This window is called the active window.

5.2.3.1 Positive Indication of the Active Window

When more than one window is open, provide a clear, positive indication to the user of the active window by means of a more complex border, subtle change in color hue, or change in labeling. This active window should be distinct, yet not distract the user's attention from window activity. See Figure 5.14.

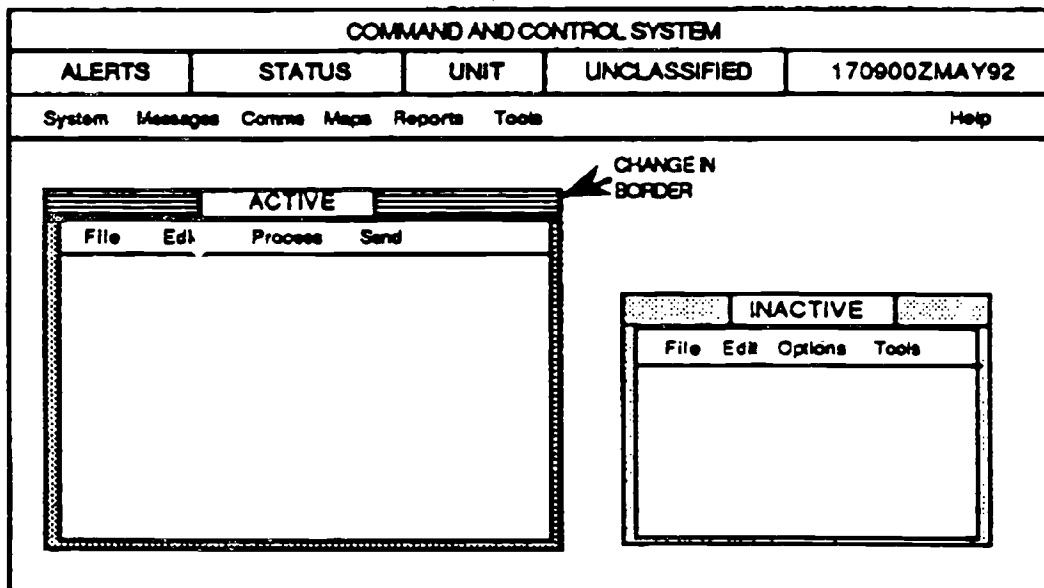


FIGURE 5.14: Example of Active Window Designation

5.2.3.2 Easy Shifting Among Windows

If several window overlays are displayed at once, provide some easy means for the user to shift among them to select which window will be currently active. For example, shift the cursor with the mouse, then press the mouse button to designate the active window.

5.2.4 LABELING

5.2.4.1 Labeling Windows

Window overlays, dialog boxes, or subordinate windows should be assigned an identifying label. This label should describe briefly the contents, purpose of the window, or the menu path. For example, Standard Message Formats.

5.2.4.2 Window Label Location

Window titles should be located at the top of the window display, with at least one blank line between the title and the body of the window.

5.2.4.3 Format of Subordinate Window Labels

Titles of subordinate windows should match the menu selection items from the supraordinate window menu.

5.2.4.4 Window Titles

Location of window titles should be consistent.

5.2.5 OPEN WINDOW NAVIGATION

Navigation, in terms of windows, refers to the user's ability to move among the various windows that are open on a display.

5.2.5.1 Open Window Map

Applications should, when using an overlapping window structure, provide a user-requested iconic or text map/indication of all open windows to allow the user to easily identify all open (especially the hidden) windows. Figure 5.15 shows three presentations of an open window map.

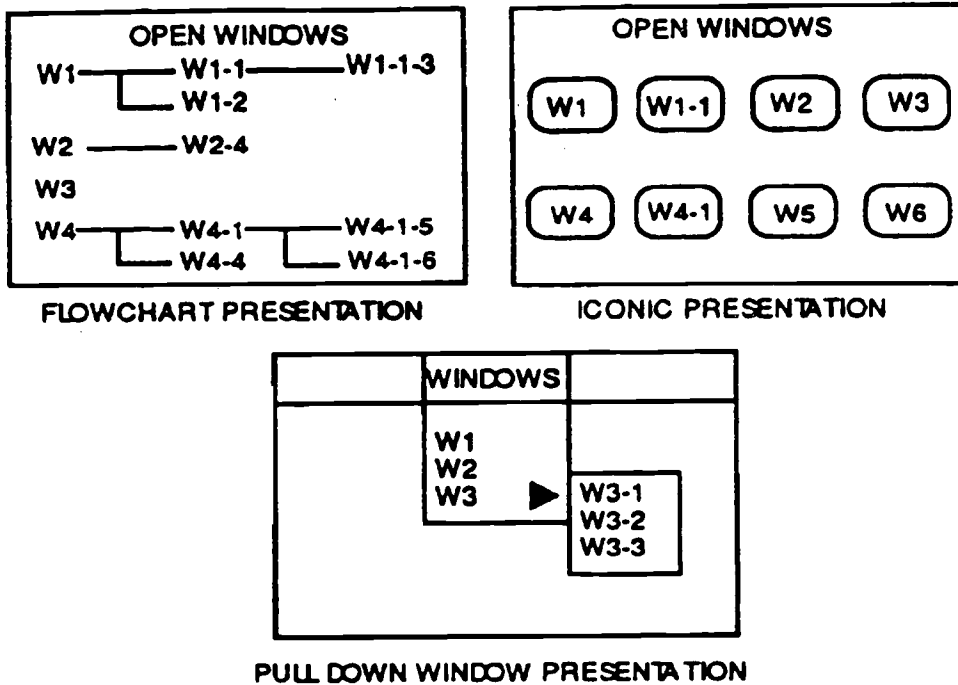


FIGURE 5.15: Examples of Open Window Maps

5.2.5.2 Active Designation from Open Window Map

Provide the user the capability to designate the active window through the iconic or text open window map by highlighting the window representation.

5.2.5.3 Expanded Window Explanation of Open Window Map

If possible, allow the user to query an open window map for expanded information (e.g., date created, size, description of subject or application, etc.) on the file or application operating in the window.

5.2.5.4 Window Forward Function With Window Map

When an iconic or text map is provided for determining the numbers and names of open windows in an overlapping system, allow the user to bring a window forward from the map without having to resize or move other windows.

REFERENCES

<u>Paragraph</u>	<u>References</u>
5.0	Avery and Bowser (1992); Billingsley (1988); DISA/CIM (1992)
5.1	DISA/CIM (1992)
5.2.1.1	Billingsley (1988) p. 416
5.2.1.2a	Avery et al. (1990); Smith and Mosier (1986) paras 2.5.2.5-3 and 2.5.8
5.2.1.2b	Avery et al. (1990); Smith and Mosier (1986) para 2.5.2.5-2
5.2.1.3a	Smith and Mosier (1986) para 2.5.2.5-1
5.2.1.3b	Slominski and Young (1988); Avery et al. (1990)
5.2.1.3c	Smith and Mosier (1986) para 2.5.2.5-10
5.2.1.4	Billingsley (1988) p. 421
5.2.1.5a	Billingsley (1988) p. 421
5.2.1.5b	Smith and Mosier (1986) para 2.5-9
5.2.1.5c	Bowser (1991) p 12; Smith and Mosier (1986) para 2.5-11
5.2.1.5d	OSF (1990) Chapter 7; Avery et al. (1990)
5.2.2.1a	Smith and Mosier (1986) para 2.5.2.5-9
5.2.2.1b	Smith and Mosier (1986) para 2.5.2.5-4
5.2.2.1c	Smith and Mosier (1986) para 2.5.2.5-5
5.2.2.1d	Nielson (1987) p. 247
5.2.2.2a	Billingsley (1988) p. 428
5.2.2.2b	OSF (1990) p. 3-4
5.2.2.3a	Billingsley (1988) p. 428
5.2.2.3b	Billingsley (1988) p. 428
5.2.2.4a	Slominski and Young (1988); Avery et al. (1990)
5.2.2.4b	Billingsley (1988) p. 429
5.2.2.4c	Billingsley (1988) p. 421; OSF (1990) p. 4-5, Avery et al. (1990)
5.2.2.4d	OSF (1990) p. 3-6
5.2.2.5a	Billingsley (1988) p. 430; OSF (1990)
5.2.2.5b	Billingsley (1988) p. 430
5.2.2.5c	Slominsky and Young (1988) p. 5; Avery et al. (1990)
5.2.3.1	Billingsley (1988) p. 431; Lewis and Fallesen (1989) p. 94; Smith and Mosier (1986) paras 2.5.2.5-7
5.2.3.2	Smith and Mosier (1986) para 2.5.2.5-8

REFERENCES (Cont.)

Paragraph

References

5.2.4.1	Smith and Mosier (1986) para 2.5.2.5-6 and 2.5-10, OSF (1990) p. 3-4
5.2.4.2	Smith and Mosier (1986) para 2.5-10
5.2.4.3	Slominski and Young (1988) p. 2, 54 p. 3-4
5.2.4.4	Slominski and Young (1988) p. 2
5.2.5.1	Billingsley (1988) p. 420
5.2.5.2	Billingsley (1988) p. 431
5.2.5.3	Billingsley (1988) p. 420
5.2.5.4	Billingsley (1988) p. 420

6.0 MENU DESIGN

Using menus as a dialog is widespread within computer systems. Menus are frequently used in conjunction with other interactive methods, such as direct manipulation.

Using menus as a dialog has advantages, a major one being that it requires little training or sophistication on the user's part. A user only needs know the meaning of each menu option, then is guided step-by-step through the operation of the system. The number of keystrokes required to access a system function may also be reduced, thereby speeding the user-to-computer transaction.

On the other hand, using menus as a dialog has disadvantages. It does not enhance retention of commands and may actually increase response time for the more experienced user. Menus may take up a large part of the display surface. In addition, for complex sequences, using menus may require an extensive menu tree structure, and the user may easily become lost navigating through a complex menu tree.

A number of different types of menuing techniques are available to the designer, including pull-down, pop-up, and sequential display. Pull-down and pop-up menus tend to be used more in direct manipulation types of dialog. Sequential display, where a control action causes another menu to overwrite the previous menu, is used more in text-based systems.

All these types of menuing techniques can be hierarchical, or branching, in nature.

The following pages provide detailed design guidelines for menus used in operational systems. To ensure a high level of user performance with menus, the designer should be aware of the following general guidelines:

Consider choosing menus when:

- tasks involve choosing among a constrained set of alternative actions
- tasks require infrequent entry of data
- the user may have little training

- the computer response is relatively fast
 - tasks require infrequently used commands
 - command sets are so large that the user is not likely to commit all commands to memory
- Design the menu tree structure broad and shallow, rather than narrow and deep. Keep the number of top-level options large, with a small number of sublevels.
 - Consider the experienced user and provide a mechanism by which the menu structure can be bypassed using a direct command.

The designer should also note that some of the guidelines discussed in the following paragraphs may be more appropriate for designing sequential display menus than for menus used in direct manipulation. The designer should use judgment regarding which approach to take.

The designer should conform to a single style of interface, such as "Motif," throughout an application. Varying interface styles confuses the user. Widgets or graphics as menu item selectors should be unique and clearly identifiable by the user.

6.1 GENERAL

6.1.1 CONSIDER RESPONSE TIME AND DISPLAY RATE

If computer response time is long, create menus with a larger number of items. If display rate is slow, create menus with fewer items to reduce display time.

6.1.2 INSTRUCTIONS AND ERROR MESSAGES

Menu instructions and error messages should be indented and placed in the same position on the screen so the user knows where to look for this information.

6.1.3 EXPLICIT OPTION DISPLAY

When entries for any particular computer transaction consist of a small set of options, show those options in a menu added to the working display, rather than require a user to remember them or to access a separate display.

6.1.4 STACKING MENU SELECTIONS

For menu selection by code entry, when a series of selections can be anticipated before the menus are displayed, permit the user to combine those selections into a single stacked entry. Stacking refers to stringing multiple commands together and executing them with one action.

6.1.5 MENUS DISTINCT FROM OTHER DISPLAYED INFORMATION

If menu options are included in a display also intended for data review and/or data entry, ensure they are distinct from other displayed information. Locate menu options consistently, and use consistent visual cues for their special function.

6.1.6 MENU BARS

Menu bars provide system functions in a bar across the top of the display screen. The following guidelines apply to menu bars.

6.1.6.1 Using Menu Bars

A menu bar is best used when screen size is small. When using large-screen displays, the distance the cursor is required to travel may be too great to be effective. Large-screen displays in this context are defined as displays intended for multiple viewers. This includes projections and theater-type displays.

6.1.6.2 Visibility of Menu Bar Options

Menu bar options, as illustrated in Figure 6.1, should remain constantly visible.

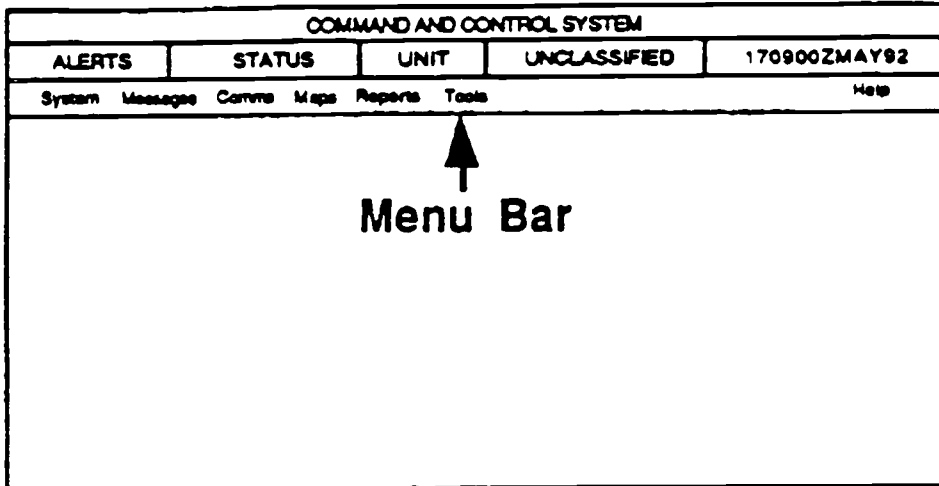


FIGURE 6.1: Example of a Menu Bar

6.1.7 PULL-DOWN MENUS

Pull-down menus, as illustrated in Figure 6.2, are lists of options attached to a selection on the menu bar that remain visible until the user takes action. Use pull-down menus instead of pop-up menus when cursor position on the screen is not important for information/option retrieval.

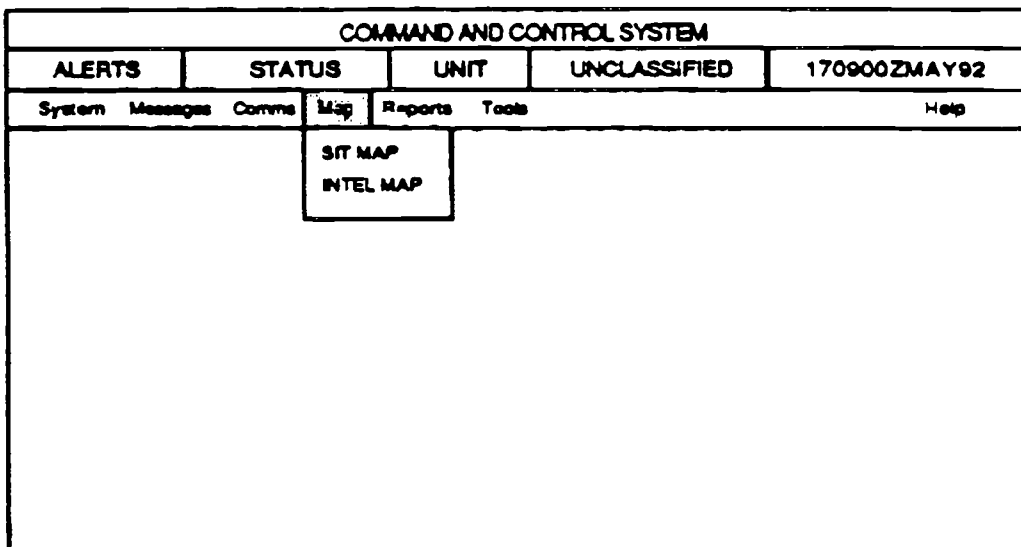


FIGURE 6.2: Example of a Pull-Down Menu

6.1.8 POP-UP MENUS

Pop-up menus are lists of options that appear on the display screen in the form of a window (see Section 5). Pop-up menus are specific to their area on the display; each window or object may have its own individual pop-up menu. The following guidelines should be used when designing pop-up windows.

6.1.8.1 Pop-Up Menu Location

Pop-up menus should be connected to the cursor location and pop up near the object or higher level menu being manipulated, as illustrated in Figure 6.3.

6.1.8.2 Selecting Options From Pop-Up Menu

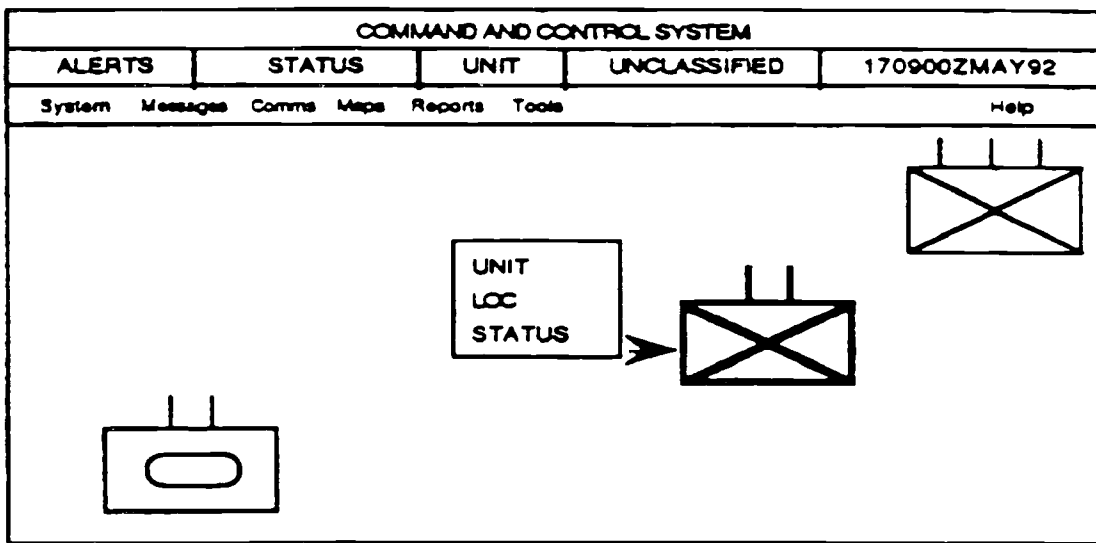


FIGURE 6.3: Example of a Pop-Up Menu

Two methods to select from a pop-up menu are: 1) hold the button down while traversing options, then release to make the selection, or 2) move the cursor and press the button again for the selection. Use the second method because, although not the easier, it is less error-prone.

6.1.8.3 Selection Highlighting

When an option has been selected from a pop-up menu, it should remain highlighted.

6.2 FORMAT

6.2.1 GENERAL

6.2.1.1 Menu Format

Lists of menu and submenu items should be brief (no more than five to nine options), arranged in separate columns, aligned, and left-justified.

6.2.1.2 Consistent Display of Menu Options

When menus are provided across different displays, design them so option lists are consistent in wording and order.

6.2.1.3 Logical Grouping of Menu Options

Format a menu to indicate logically related groups of options, rather than an undifferentiated string of alternatives.

6.2.1.4 Logical Ordering of Grouped Options

If menu options are grouped in logical subunits, display those groups in a logical order. If no logical structure is apparent, display the groups in the order of their expected frequency of use. See the example in Figure 6.4.

6.2.1.5 Sequence or Frequency Ordering

For a small number of menu items, use sequence or frequency to determine menu order.

6.2.1.6 Alphabetic Ordering

For a large number of menu options, use alphabetic ordering of menu items.

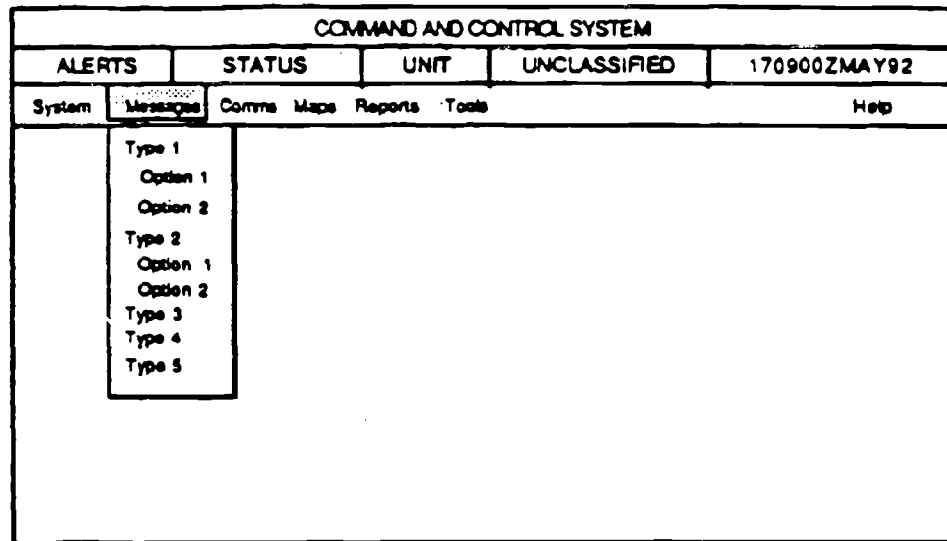


FIGURE 6.4: Example of Logical Ordering of Grouped Options

6.2.1.7 Numbering Menu Options

When task order is important, menu options should be listed by number, not by letter.

6.2.1.8 Display of Options

Design of a menu for a GUI should display unavailable menu items in a visually distinct manner. Refer to Paragraph 8.3.3.5.

6.2.1.9 Single-Column List Format

When multiple menu options are displayed in a list, display each option on a new line (i.e., format the list as a single column).

6.2.1.10 Overlapping Items

Ensure menu options do not overlap controlled functions or appear to do so to the user.

6.3 HIERARCHICAL MENUS

6.3.1 USAGE

Use hierarchical menus:

- a. when menu selection must be made from a long list and not all options can be displayed at once
- b. if a selection list exceeds 10-15 items.

6.3.2 GENERAL GUIDANCE

6.3.2.1 Organization and Labeling of Hierarchical Menus

When hierarchical menus are used, organize and label them to guide the user within the hierarchical structure. Currently active menu selections need to be identified to the user. The preferred method would use more than one mode (i.e., color and font, size and color of text, etc.).

6.3.2.2 Easy Selection of Important Options

Design hierarchical menus to permit immediate user access to critical or frequently selected options.

6.3.2.3 Indicating Current Position in Menu Structure

When hierarchical menus are used, display an indication of the user's current position in the menu structure. This could be done in the menu title, or as a page X of N notation on the menu page.

6.3.2.4 Consistent Design of Hierarchical Menus

When hierarchical menus are used, ensure the display format and option selection logic are consistent at every level of the hierarchical menu structure.

6.3.2.5 Graphic User Interface for Hierarchical Menus

Hierarchical menu design in a GUI should be as simple as possible. The use of complex graphic structures is distracting to the user.

6.3.3 NAVIGATING HIERARCHICAL MENUS

6.3.3.1 Including a System-Level Menu

Provide a system-level menu of basic options as the top level in a hierarchical menu structure, as illustrated in Figure 6.5. The system-level menu will act as a home base to which a user can always return as a consistent starting point for control entries.

6.3.3.2 Organization and Labeling of System-Level Menu Listed Options

COMMAND AND CONTROL SYSTEM					
ALERTS	STATUS	UNIT	UNCLASSIFIED	170900ZMAY92	
System	Messages	Comms	Maps	Reports	Tools
					Help

FIGURE 6.5: Example of a System-Level Menu

Control options for the system-level menu should be grouped, labeled, and ordered in terms of their logical function, frequency, and criticality of use.

6.3.3.3 Return to the System-Level Menu

When hierarchical menus are used, require the user to take only one simple control action to return to the system-level menu.

6.3.3.4 Return to Higher Level Menus

When hierarchical menus are used, require the user to take only one simple control action to return to the next higher level.

6.3.3.5 Control Options Distinct From Menu Branching

Format the display of hierarchical menus, dialog boxes, and pop-up windows so that options actually accomplish control entries can be distinguished from those which merely branch to other menu frames. See Figure 6.6.

6.3.3.6 Hierarchical Menu-Browsing Methods in Direct Manipulation

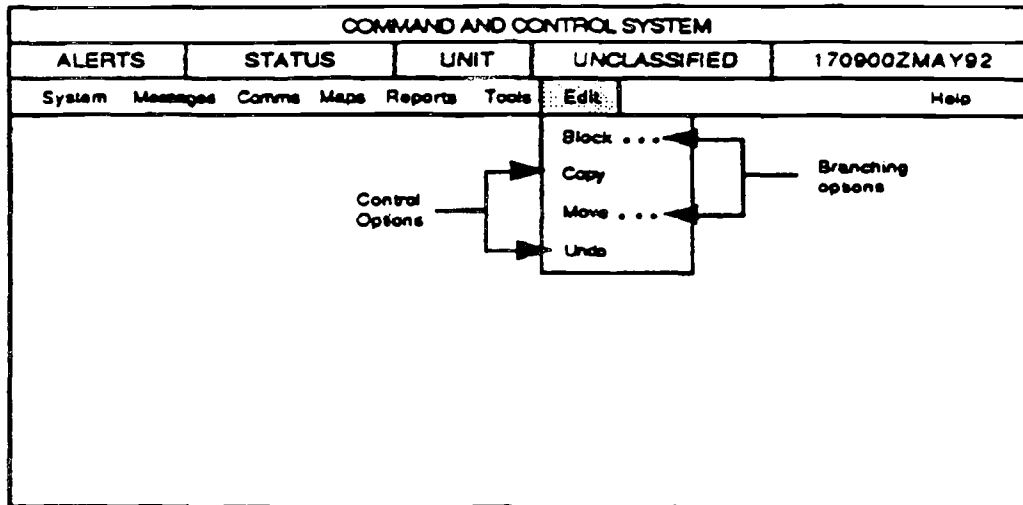


FIGURE 6.6: Distinction Between Control Options and Command Options

Two basic methods for browsing options in hierarchical menus are used in direct manipulation interactive control: 1) select an option from one menu, which causes another menu to pop up, or 2) move the cursor towards the right side of an option, causing a menu to pop up. See Figure 6.7.

6.3.3.7 Use of Multiple Paths

Provide multiple paths to accommodate both the experienced and inexperienced user. Allow the experienced user to use "type-ahead," "jump-ahead," or other shortcuts to navigate through the menu selection system.

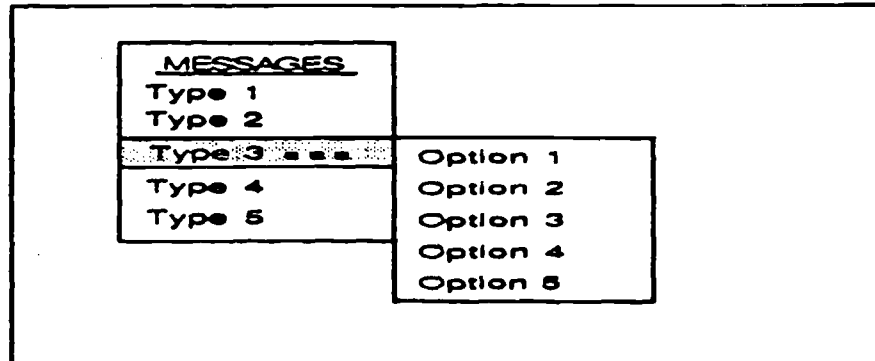


Figure 6.7: Example of a Hierarchical Menu

6.3.4 HIERARCHICAL MENU TREE DEPTH AND BREADTH

6.3.4.1 Minimal Steps in Sequential Menu Selection

When the user must step through a sequence of menus to make a selection, design the hierarchical menu structure to minimize the number of steps required.

6.3.4.2 Use Broad Menu Trees

Use a broad and shallow menu tree, rather than a narrow and deep menu tree for operational systems, as illustrated in Figure 6.8.

6.3.4.3 Minimize Menu Choices in the Middle

Minimize the number of menu choices midway through a hierarchical menu, as the user is more likely to get lost at this stage.

6.3.4.4 Software Navigation Aids

Software navigation aids should include the ability to select a menu or submenu directly, without going through intermediate steps (Figure 6.9).

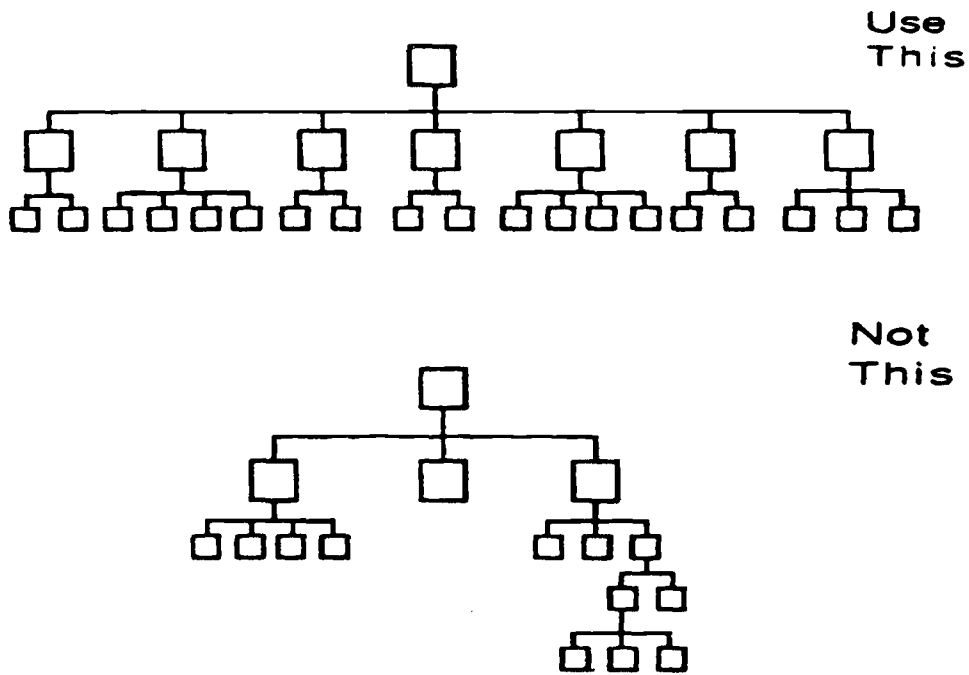


FIGURE 6.8: Broad and Shallow Menu Tree vs. Narrow and Deep Menu Tree

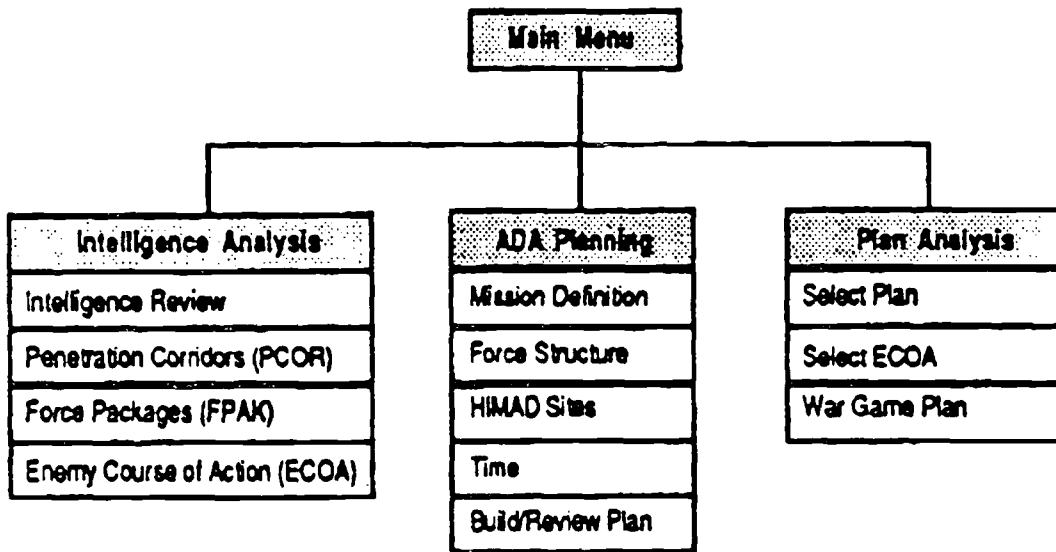


FIGURE 6.9: Example of a Tree Diagram Interface

6.4 ITEM SELECTION

6.4.1 GENERAL

6.4.1.1 Automatic Cursor Placement

When pointing to make a menu selection, on menu displays not included with data displays, the computer should place the cursor automatically at the first listed option. When menu selection is by code entry, place the cursor in the command entry area.

6.4.1.2 Minimize Menu Selections

Keep the number of menu selections to the absolute minimum to reduce system menu-selection time.

6.4.1.3 Use a Combined Mode of User Interface

Users should be able to use two modes for menu selection: keying in a numeric or letter code, or placing the cursor at the option and selecting. See Figure 6.10.

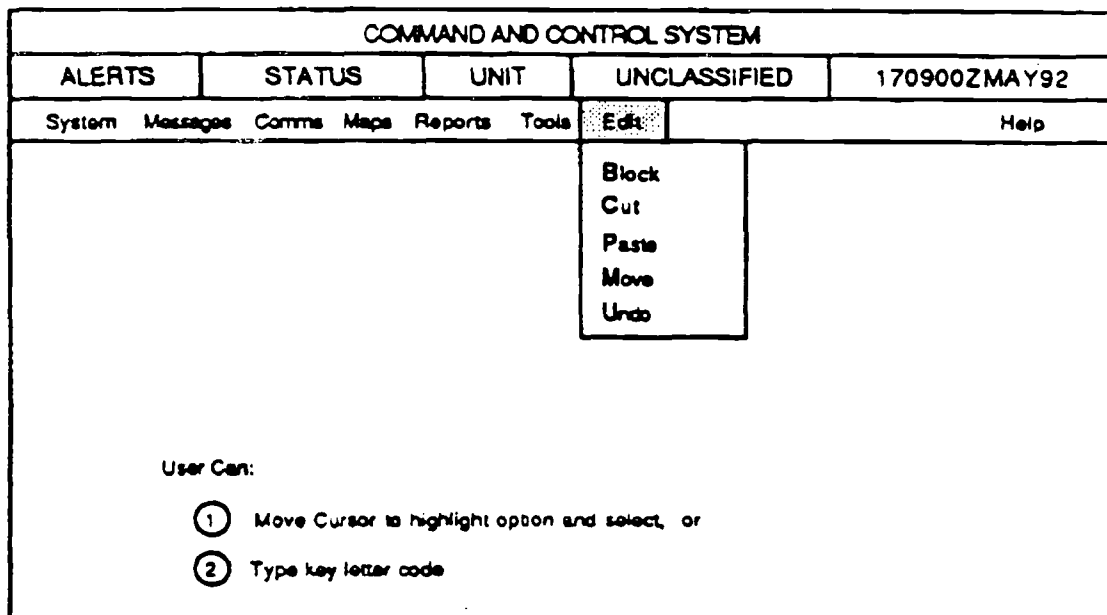


FIGURE 6.10: Example of a Combined Mode User Interface

6.4.1.4 Feedback for Menu Selection

When a user selects and enters a control option from a menu, if no natural response is immediately observable, the software should display some other acknowledgment of that entry. See examples in Figure 6.11.

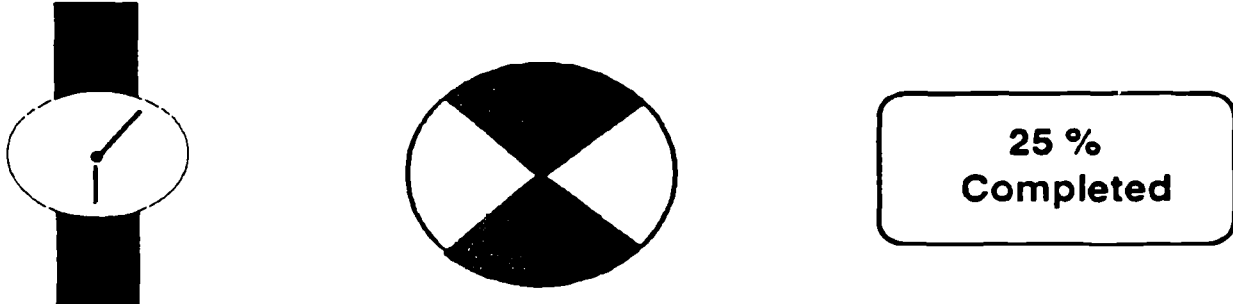


FIGURE 6.11: Icons that Acknowledge Entry Selection

6.4.1.5 Standard Area for Code Entry

When menu selection is accomplished by code entry, provide a standard command entry area where the user enters the selected code. Place that entry area in a fixed location on all displays.

6.4.1.6 Allow Abbreviated Menu Selections

Allow menu selections by the user to be accepted in either abbreviated or complete form. For example, the user should be able to use Q, QU, or QUIT.

6.4.2 SELECTION BY POINTING

6.4.2.1 Menu Selection by Pointing

If menu selection is the primary means of sequence control, and especially if choices must be made from extensive lists of displayed control options, permit option selection by direct pointing (e.g., mouse, trackball, etc.). See Section 7.1, Direct Manipulation.

6.4.2.2 Large Pointing Area for Selecting Options

The acceptable pointing area for menu options should be as large as is consistently possible. The area should include at least the displayed option label, plus a half-character distance around that label.

6.4.2.3 Dual Activation for Pointing

If menu selection is accomplished by pointing, provide dual activation, where the first action designates (positions a cursor at) the selected option, and a separate, second action makes an explicit control entry (e.g., clicking the mouse).

6.5 MENU OPTION LABELING

6.5.1 GENERAL

6.5.1.1 Use of Key Words

Menu items should begin with a key word.

6.5.1.2 Menu Options Worded as Commands

The wording of menu options should consistently represent commands to the computer, rather than questions to the user (e.g., File, Save, Edit, etc.).

6.5.1.3 Menu Categories

Menu category labels should be comprehensible and unique. The words, phrases, and titles should state options in clear English.

6.5.1.4 Labeling Grouped Options

If menu options are grouped in logical subunits, give each group a descriptive label that is distinctive in format from the option labels themselves. See Figure 6.12.

6.5.1.5 Use Familiar Terminology

Use familiar terminology when labeling menus, but ensure items are distinct from one another.

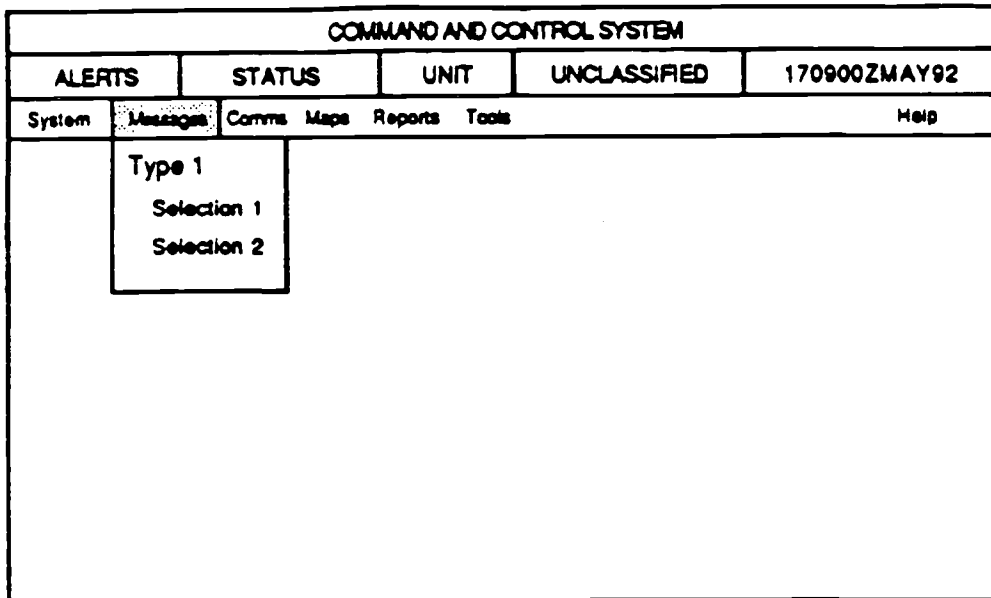


FIGURE 6.12: Example of Distinctive Subunit Labels

6.5.2 SELECTOR

6.5.2.1 Best and Worst Selectors for Menu Items

Mnemonics is a technique to assist in improving the user's memory.

Compatible or mnemonic letters are the best selectors for menu items, while incompatible letters are the worst. Numbers are intermediate selectors.

- a. Lettered menu items have the following advantages: more single entry keys are available; there is less chance of a keying error; and mnemonic keying of entries is possible.
- b. Numbered menu items have the following advantages: sequencing of items is clear; non-typists can easily locate numbers; and the user can quickly see how many options are available.

6.5.2.2 Do Not Combine Codes

Letter and numeric codes should not be combined in the dialog.

6.5.2.3 Selection of Menu Titles

Use selectors that closely match the item represented, to facilitate user retention of commands.

6.5.2.4 Numbering

Number menu items starting with 1 -- not with 0.

6.5.2.5 Consistent Coding of Menu Options

If letter codes are used for menu selection, use those letters consistently in designating options from one transaction to another.

6.5.2.6 Displaying Option Code

When the user must select options by code entry, display the code associated with each option in a consistent, distinctive manner. See Table 6.1.

TABLE 6.1: Examples of Option Codes

<u>Code</u>	<u>Option</u>
P =	Previous Page
N =	Next Page
U =	Undo
Del =	Delete

6.6 DIALOG BOXES/POP-UP WINDOWS

GUI style guides refer to windows that contain graphical controls, such as dialog boxes and pop-up windows, for interacting with applications. Examples of dialog boxes include message, question, warning, action, and command windows; examples of pop-up windows include command windows, property windows, and notices. Note that dialog box and pop-up command windows are not equivalent or even related. The former refers to a window that allows users

to enter commands to the application or operating system, and the latter is a window that sets parameters and executes commands based on those parameters.

These windows are used to:

- display important messages or warnings
- collect or solicit data from the user
- modify and set properties of objects
- notify the user of the progress of a lengthy process.

Dialog boxes and pop-up windows are invoked by applications in response to 1) user actions and requests, 2) unexpected or unplanned events (e.g., a printer runs out of paper) or 3) initiation of a time-consuming activity. The application decides where and when they are displayed, but all dialog boxes and pop-up windows should include at least one button that solicits a response from the user. They should be noticeable but small, and, if possible, moveable. It is recommended that only one dialog box or pop-up window should be displayed at a time within any application in order to avoid clutter and confusion.

Dialog boxes and pop-up windows should automatically receive input focus. Users should be required to respond to dialog boxes or pop-up windows, users should be prevented from returning the input focus to the main or primary window (of the application) until they have responded correctly.

All types of dialog boxes and pop-up windows behave similarly, but they differ in content depending on the needs of the application. For example, a push button is always pushed and a check button is always checked, but each application will choose the types of controls to use and combine them differently. The following sections provide recommendations for message wording and briefly describe some common types of dialog boxes and pop-up windows. More detailed descriptions can be found in the OSF/Motif and Open Look style guides.

6.6.1 MESSAGE WORDING GUIDELINES

The following guidelines, which are designed to maximize user performance and accuracy, should be applied to dialog boxes, pop-up windows, message areas, and any other communications between the application and user.

- Use an abbreviation only when it is significantly shorter than the full word.

- Ensure abbreviations are meaningful, recognizable, and used consistently.
- Do not abbreviate words not commonly abbreviated. For example, use "Restricted Acct No," not "Restr Account Number."
- Ensure message lines end in full words rather than hyphenations.
- Ensure messages are directly usable, requiring no further documentation or translation.
- Avoid overly technical wording and use short simple sentences that begin with the main topic.
- Avoid abrupt wording such as INVALID, ILLEGAL, and FATAL.
- Focus error messages on the procedure for correcting the error, not the action that caused the error.
- Display critical error messages (those that require immediate response from the user to prevent invalid data or results) in caution/warning windows, as shown in Section 6.6.4. Display noncritical messages in the message area at the bottom of the application window, as described in Section 5.1.5.

6.6.2 WORK-IN-PROGRESS WINDOW

When a user's request is simple and requires five seconds or less processing time, feedback can be in the form of a changed pointer shape or a brief message within the window. When the request five seconds, the application should provide a work-in-progress window to indicate a time-consuming operation is taking place. If appropriate, provide a means by which the operation can be canceled or aborted. The application removes the box when the operation has been completed.

The application should show the status of the operation by a dynamically changing progress indicator (e.g., "10% Sorted," "4 out of 10 files copied," or a scale showing status).

6.6.3 INFORMATION BOX

An application should generate an information box (i.e., a Motif message box or an Open Look notice) when the application needs to display an information message. This window should be reserved for noncritical messages requiring

acknowledgement by the user. An application's frequent informational messages should be displayed in the window's message area (see Section 5.1.5).

An information box can freeze the application and require the user to explicitly dismiss the window before proceeding. If the halted operation can be retried, include a "Retry" button within the message window. If a default push button is designated, assume it is the desired action.

6.6.4 CAUTION/WARNING BOX

A caution/warning box contains critical messages that warn the user of the consequences of carrying out an action and usually includes "Yes," "No," and "Cancel" buttons. The message should be an unambiguous question or statement. When this box is displayed, the application is suspended until the user provides instructions on how to proceed. The default push button should always be the least destructive operation.

6.6.5 MENU BOX

A menu box is the result of the user's selecting a routing or window menu item. Menu boxes solicit data from users through a combination of controls (e.g., entry boxes and settings). The menu box should be named in accordance with the menu item that created it. For example, the "search..." menu item should generate a menu with the title "Search....". A "Cancel" push button should be included in the window to allow users to dismiss the menu box. If a default push button is designated, it should be the assumed desired action.

REFERENCE LIST

<u>Paragraph</u>	<u>References</u>
6.1.1	Shneiderman (1987) p. 107
6.1.2	Shneiderman (1987) p. 115
6.1.3	Smith and Mosier (1986) para 3.1.3-16; Sidorsky (1984) para 6.1-14
6.1.4	Smith and Mosier (1986) para 3.1.3-36
6.1.5	Smith and Mosier (1986) para 3.1.3-20
6.1.6.1	Ziegler and Fähnrich (1988) p. 130
6.1.6.2	Ziegler and Fähnrich (1988) p. 130
6.1.7	Ziegler and Fähnrich (1988) p. 130
6.1.8.1	Ziegler and Fähnrich (1988) p. 129
6.1.8.2	Ziegler and Fähnrich (1988) p. 129
6.1.8.3	Ziegler and Fähnrich (1988) p. 129
6.2.1.1	Lickteig (1989) p. 13, 30 Appendix A p. A-1; Bailey (1982) p. 346; Chao (1986) p. 15
6.2.1.2	Smith and Mosier (1986) para 3.1.3-19; Shneiderman (1988) p. 702.
6.2.1.3	Smith and Mosier (1986) para 3.1.3-22 and 4.4-3; Shneiderman (1988) p.702; Paap and Roske-Hofstrand (1986) p. 384
6.2.1.4	Smith and Mosier (1986) para 3.1.3-23; Lickteig (1989) p. 15; Shneiderman (1988) p. 702
6.2.1.5	Galitz (1984) p. 120
6.2.1.6	Galitz (1984) p. 120; Chao (1986) p. 16
6.2.1.7	Williams et al. (1987a) Appendix p. A-3
6.2.1.8	Smith and Mosier (1986) para 3.1.3-17 and 3.1.3-18; Sidorsky (1984) para 6.1-13
6.2.1.9	Smith and Mosier (1986) para 3.1.3-3
6.2.1.10	Shneiderman (1987) p. 100
6.3.1a	Smith and Mosier (1986) para 3.1.3-25; Nielsen (1987) p. 384; Lickteig (1989) p. 13; Ziegler and Fähnrich (1988) p. 129
6.3.1b	Sidorsky (1984) para 6.1-14; Chao (1986) p. 15
6.3.2.1	Bowser (1991) p. 9; Smith and Mosier (1986) para 4.4-4
6.3.2.2	Smith and Mosier (1986) para 3.1.3-28; Sidorsky (1984) para 5.1-14; Galitz (1984) p. 119
6.3.2.3	Smith and Mosier (1986) para 3.1.3-30; DoD (1989a) p. 267; Chao (1986) p. 15; Shneiderman (1987) p. 115
6.3.2.4	Smith and Mosier (1986) para 3.1.3-32

REFERENCE LIST (Cont.)

<u>Paragraph</u>	<u>References</u>
6.3.2.5	Bowser (1991) p. 9
6.3.3.1	Smith and Mosier (1986) para 3.1.3-26, 4.4-2, 3.2-2; Shneiderman (1988) p. 702; Galitz (1984) p. 120; Sidorsky (1984) para 1.1-12, 3.2-21
6.3.3.2	Smith and Mosier (1986) para 3.2-3
6.3.3.3	Smith and Mosier (1986) para 3.1.3-34; Galitz (1984) p. 120; DoD (1989a) p. 266
6.3.3.4	Bowser (1991) p. 10; Smith and Mosier (1986) para 3.1.3-33; Sidorsky (1984) para 5.1-14; DoD (1989a) p. 267; Chao (1986) p.16
6.3.3.5	Smith and Mosier (1986) para 3.1.3-31; Galitz (1984) p. 120
6.3.3.6	Ziegler and Fähnrich (1988) p. 129
6.3.3.7	Smith and Mosier (1986) para 3.1.3-35; Shneiderman (1988) p.702; Shneiderman (1987) p. 118; Chao (1986) p. 16; Laverson and Shneiderman (1987) p. 104; Sidorsky (1984) para 5.1-15
6.3.4.1	Smith and Mosier (1986) para 3.1.3-27; Sidorsky (1984) para 5.1-14
6.3.4.2	Shneiderman (1988) p. 702; Norman and Chin (1988) p. 63
6.3.4.3	Norman and Chin (1988) p. 63
6.3.4.4	Galitz (1984) p. 120
6.4.1.1	Galitz (1984) para 3.1.3-29; Chao (1986) p. 15
6.4.1.2	Parkinson et al. (1988) p. 691
6.4.1.3	Antin (1988) p. 181
6.4.1.4	Smith and Mosier (1986) para 3.1.3-9
6.4.1.5	Smith and Mosier (1986) para 3.1.3-8; Chao (1986) p. 15
6.4.1.6	Sidorsky (1984) para 6.1-15
6.4.2.1	Smith and Mosier (1986) para 3.1.3-4; DoD (1989a) p. 266
6.4.2.2	Smith and Mosier (1986) para 3.1.3-5
6.4.2.3	Smith and Mosier (1986) para 3.1.3-6
6.6.1.1	Shneiderman (1988) p. 702
5.5.1.2	Smith and Mosier (1986) para 3.1.3-11; Sidorsky (1984) para 5.1-13
6.5.1.3	Shneiderman (1987) p. 87; Bailey (1982) p. 343
6.5.1.4	Smith and Mosier (1986) para 3.1.3-24; Sidorsky (1984) para 5.1-14; Chao (1986) p.16
6.5.1.5	Shneiderman (1987) p. 100

REFERENCE LIST (Cont.)

Paragraph

References

- | | |
|---------|--|
| 6.5.2.1 | Laverson et al. (1987) p. 106; Shneiderman (1987) p. 117 and 118; Chao (1986) p. 16; Smith and Mosier (1986) para 3.1.3-13; Galitz (1984) p. 120 |
| 6.5.2.2 | Chao (1986) p. 16 |
| 6.5.2.3 | Laverson et al. (1987) p. 105 |
| 6.5.2.4 | Sidorsky (1984) para 2.1-13; Chao (1986) p. 16 |
| 6.5.2.5 | Smith and Mosier (1986) para 3.1.3-14; Sidorsky para 5.1-14 |
| 6.5.2.6 | Smith and Mosier (1986) para 3.2-8; Shneiderman (1987) p. 115 |
| 6.6 | DISA/CIM (1992) |

7.0 OBJECT ORIENTATION

For GUIs, a major type of interactive dialog is through direct manipulation. In a direct manipulation dialog, the user controls the interface with the computer by acting directly on "objects" on the display screen. An object may be an icon (which is a pictographic symbol that represents a computer process or data [Gittins 1986]), menu option, symbol, button, or dialog box. The user highlights the object and implements the action by the use of a pointing device, such as a mouse. Actions include moving an object, querying a database, calling up a preformatted message, or sending a message over a communications system. The result of an action is immediately observable.

The direct manipulation of a computer system is analogous to controlling a tactical vehicle. The user uses a control, such as the steering wheel, to input a command to the vehicle and is rewarded by an immediate response. With a computer, the user moves a cursor over an object, such as a symbol, and presses a button to input a command, such as querying status. The system responds immediately by displaying the status in a pop-up window next to the object. This is unlike a command-language-based system, where the user types in a command, hits ENTER, then waits for a response from the system.

Direct manipulation user interfaces are characterized by continuous representation of the object of interest, physical actions or button presses to effect computer actions, incremental reversible actions, and immediate visual feedback. These characteristics provide the user with a greater feeling of control and often lead to better performance and greater acceptance.

A reason for using direct manipulation in the user interface is the reduced time required to learn new applications. Efficiencies in learning result from using both standard, consistent actions in the application environment and metaphors to guide the user. A metaphor uses the visual nature of a direct manipulation interface to map objects in the application onto a visual representation familiar to the user. Metaphors effectively control the complexity of the user interface by making actions, procedures, and concepts similar to those already known to the user. By exploiting the user's prior knowledge, the designer permits the user to think in terms familiar to the application domain, rather than in terms of low-level computer concepts. This results in applications that are easier to learn and easier to use. Some effective examples are an accountant's

worksheet in a spreadsheet application or the physical world in an object-oriented programming environment.

Note the following considerations when designing an object-oriented, user-computer interface:

- Selecting the metaphor should leverage prior knowledge in a way that is specific to the user's environment. The metaphor should represent the system function in a way that is meaningful to the user. The office metaphor described above may not be appropriate for a tactical military system.
- The metaphor should be simple, but not so simple that it leads to underutilization of the system because it does not model all the available capability. The user will not be aware of what is not modeled.
- The metaphor used for icons and system interaction should be tested with representatives of the intended user population.
- The following hardware considerations should be incorporated for an effective direct manipulation system: 1) a high-resolution, bit-mapped display system, 2) greater central processing and memory to support the bit-mapped windowing system, and 3) rapid operating speed to provide the immediate response needed for effective user-computer interaction.

The designer should recognize that the ways in which a human perceives figures impacts how icons are designed. The icon designer needs to incorporate Gestalt principles of human perception (Lewis and Fallesen 1989), briefly described as follows:

- Humans see the simplest or most efficient interpretation of an icon. Therefore, the designer should avoid complex shapes that the user may misinterpret.
- The user will associate a meaning with an icon. The stronger the associated meaning, the more easily the icon will be recognized and remembered. Therefore, the designer should create icons that have intrinsic meaning to the user.
- Users tend to mentally group objects. Grouping can be based on proximity, similarity, arrangement of objects that define a closed region, arrangement of objects in straight or smoothly curving lines, symmetry of arrangement, and objects undergoing simultaneous, correlated changes. Grouping can provide

additional meaning to icons, but the designer should ensure that unrelated icons are not inadvertently grouped by users.

Figure-ground relationships are important to how a user perceives an icon. Figure refers to an object, which has a shape and stands out from the background. Ground refers to the area that is perceived to continue behind the figure. Keep in mind the following points regarding this topic:

- The size of a figure relative to its background is important. The smaller the size of a figure relative to the ground, the more likely it will be perceived as a figure.
- When shape only is used as a discriminator for figures, convex shapes will likely be perceived as figures and concave shapes as holes.
- A contour line will be perceived as belonging to only one of the areas it delineates.
- Position affects whether an object will be perceived as a figure. Centrally positioned objects and the lower portion of a surface divided horizontally into two parts are seen as figures.
- The greater the contrast between an object and its background, the greater the perception of the object as a figure.

The following pages provide detailed guidelines for designing the user-computer interface when direct manipulation is used. The designer should recognize that the literature has little explicit guidance for direct manipulation as it applies to specific military systems. Therefore, it is imperative that before formalizing the user-computer interface, detailed research and careful testing of alternatives be done with users who represent the intended user population.

7.1 DIRECT MANIPULATION

7.1.1 GENERAL

Provide direct manipulation of displayed objects as a means of interactive control, because operational military applications may be used by casual system users, and turnover in personnel will be high.

7.1.1.1 Hardware Considerations

The designer should be aware of the following hardware considerations:

- a. High-resolution screens and a bit-mapped hardware architecture are required for direct manipulation systems.
- b. Because direct manipulation is designed to represent the actual product, a positive image (dark foreground on light background) is best, as it represents printed output. See the example in Figure 7.1.

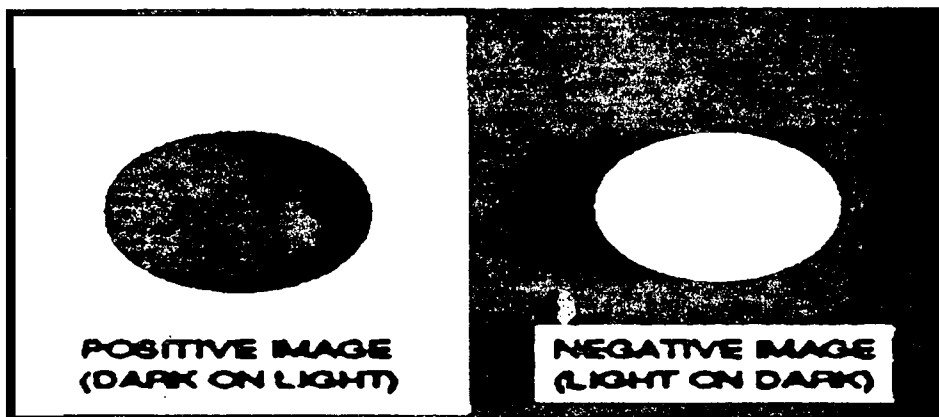


FIGURE 7.1: Example of Positive and Negative Images

- c. Direct manipulation is most efficient when a pointing device such as a mouse, trackball, or other device is used. Touch-interactive devices are discussed in Section 3. Software must be flexible enough to accommodate keyboard cursor keys should the pointing device fail.

7.1.1.2 Screen Arrangement by the User

The user should be able to arrange windows and icons on the screen to meet the individual task needs. However, the user should not be able to move the window or icon off the screen.

7.1.1.3 Function Control

Five methods should be considered for invoking a function, file, or operation with direct manipulation: Function Keys, Menu Bar, Pop-up Menus, Pull-down Menus, and Icons. Icons are discussed in Paragraph 7.1.3. Function keys are discussed in Section 8.4, and menuing is discussed in Section 6.

7.1.1.4 Interaction

Operator interactive tasks should use the most appropriate input mode. The keyboard is recommended for extensive alphanumeric data. Using pointing devices to select from menus is usually more effective. Where both modes are present, the software should allow both keyboard and pointing device selection of items.

7.1.2 METAPHORS

Metaphors associate interface objects in the application or problem domain with a visual representation familiar to the user (e.g., trash cans equal delete). They should be used carefully and tested with the user population. The following paragraphs provide guidance on metaphor design.

7.1.2.1 Metaphor Model

The metaphor should:

- a. model the system being controlled
- b. be appropriate for the user's tasks, functions, and environment. The office metaphor may not necessarily be appropriate for tactical military applications.

7.1.2.2 Metaphor Consistency with Objects

Metaphors should be consistent with the objects chosen to represent the functions. For example, deleting a file with recovery capability would be represented by a trash can, whereas deleting a file permanently would be represented by a paper shredder.

7.1.2.3 Complex Metaphors

Avoid using complex metaphors. Complex metaphors, like complex icons, can lead to increased inferences of meaning and errors by the user.

7.1.2.4 Metaphor Oversimplification

Although metaphors should be as simple as possible, avoid oversimplification. Oversimplification occurs when the metaphor does not model all the capability available to the user, causing under-utilization of the system.

7.1.2.5 Metaphors Versus Self-Contained Icons

If effective self-contained symbols (icons) can be designed for information presentation, use them over multiple icons of a complex metaphor.

7.1.2.6 Generalization of Metaphors

Metaphors should be general enough to allow the user to understand and use other metaphors or media, such as text-based systems.

7.1.2.7 Metaphor Tutoring

Design icon metaphors to tutor the user towards a more complete understanding of the underlying functional system.

7.1.2.8 Connotations Induced by Metaphors

Develop metaphors carefully, especially those used by more than one cultural or national group (e.g., NATO forces), to ensure they do not have a negative connotation for the user. See also Paragraph 7.1.3.9c.

7.1.3 ICONS

Icons are pictographic symbols that represent objects, processes, or data. Icons are the most visible manifestation of a metaphor, as described in the introduction to Section 7. The following paragraphs provide guidelines for designing icons. However, it must be stressed that more research needs to be performed to arrive at a complete set of icons for tactical military systems. The designer should consider the display media when designing the icon (e.g., an icon design for a high-resolution display may lose meaning when presented on a low-resolution display).

7.1.3.1 Use of Icons

When using direct manipulation, use icons as visual representations of system functions that can be manipulated by the user.

7.1.3.2 Iconic Menus

Iconic menus are groups of icons that act the same as textual menus, allowing selection of system options. Figure 7.2 illustrates an iconic menu.

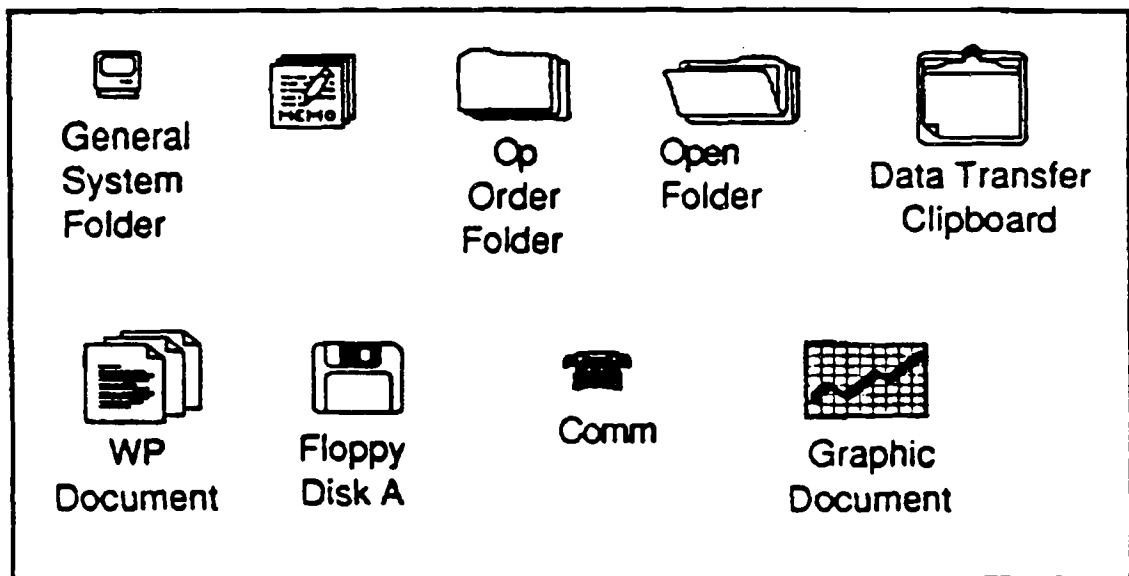


FIGURE 7.2: Example of an Iconic Menu

- a. When a user will be multilingual (e.g., NATO Forces), devise iconic menus for control functions.
- b. When using iconic menus, design the system such that once an action has been initiated through an icon (e.g., printing), nonselectable icons cannot be manipulated. Provide the user with a visual indication of which icons are unavailable.

7.1.3.3 Icon Mapping to Functions

When an icon is used to represent more than one function (i.e., mapped), do not repeat the icon for each function. Selection of the icon should cause a menu to appear that allows the user to select the specific function to be performed. For example, an icon for selecting communication devices, when selected, would cause a menu listing each type of communication device to appear. The user would then select the device to be used.

7.1.3.4 Switching to Textual Representation

Include a feature that allows the user, when working with icons, to switch to a textual representation of the functions or files. The text should be listed sequentially to produce a logical transition from icon to text. See Figure 7.3.

7.1.3.5 Testing Icon Design

It is imperative to test icon design with a group of users who represent the intended user. This should ensure the meaning of the icon is implicitly understood.

7.1.3.6 Using Color-Coding in Icons

Use caution when color-coding, and use color only if it is redundant to another coding method. See Section 4.3, Color, for additional color guidelines.

7.1.3.7 Consistency of Icon Design

- a. Icon meaning should be consistent across displays and standardized within an application. To the extent possible, it should also be standardized and consistent across all DoD applications.

- b. Icon labels should be placed underneath the icon, as illustrated in Figure 7.4. If labels are not used, the user should be able to query the system to get a definition of the icon.

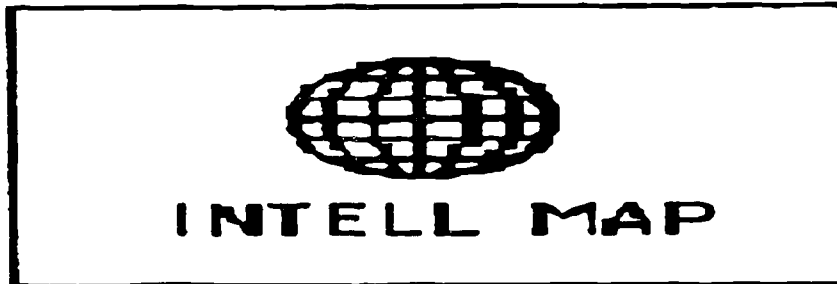


FIGURE 7.4: Example of a Text Title for an Icon

7.1.3.9 Icon Shape

- a. Design icon shapes to provide a visual representation that matches the user's expectations and allows an association between the icon and the function being controlled. Icon shape should be a concrete, not abstract, concept with respect to the user.
- b. Keep icon shapes as simple as possible to ensure user recognition. If the icon shape is too complex, the user may make errors in recognizing the icon. Icon shape should show or exaggerate an object's natural features. See Figure 7.5.

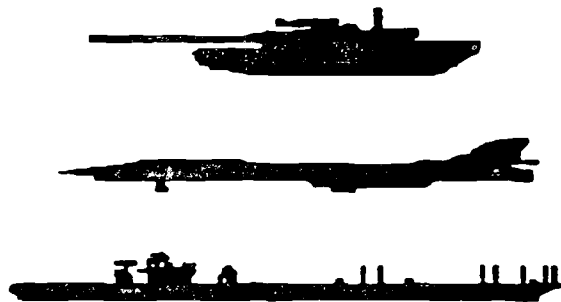


FIGURE 7.5: Examples of Icon Shapes

- c. If icons are to be used with different cultural or national groups (e.g., NATO Forces), use technological shapes or forms rather than natural objects. The examples shown in Figure 7.6 include areas shaped as circles, triangles, or boundaries shaped as lines with pluse signs (+) as end marks.

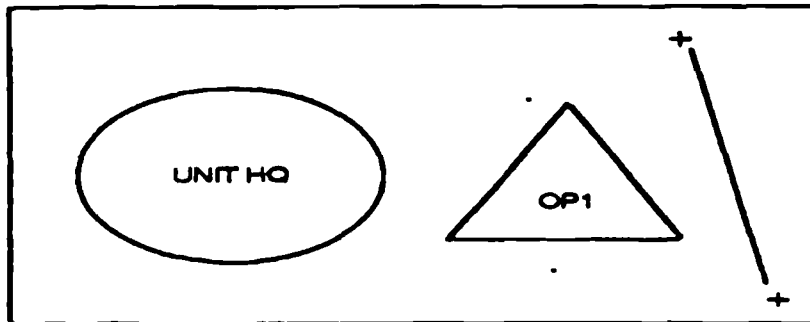


FIGURE 7.6: Example of the Use of Technological Shapes

- d. The fewer unique icon shapes used, the more effective the user-computer interface. At a maximum, no more than 20 unique shapes should be used.
- e. Icon boundary lines should be solid and closed and should have a high contrast value. Corners should be smooth. The best boundary is based on the contrast between the figure and the underlying display background.
- f. If a boundary is left open, be aware that the user will tend mentally to close the open boundary. Design the opening to ensure only desired closures will occur.
- g. Icon figure-ground (foreground lines, etc.) should be clear and stable.
- h. When icons represent opposite functions, design the icons so they mirror one another. See Figure 7.7.
- i. If three-dimensional icons are used, allow the user to manipulate them through rotation so they can be viewed from different vertical and horizontal viewing angles.
- j. When an icon is selected, it should be highlighted.

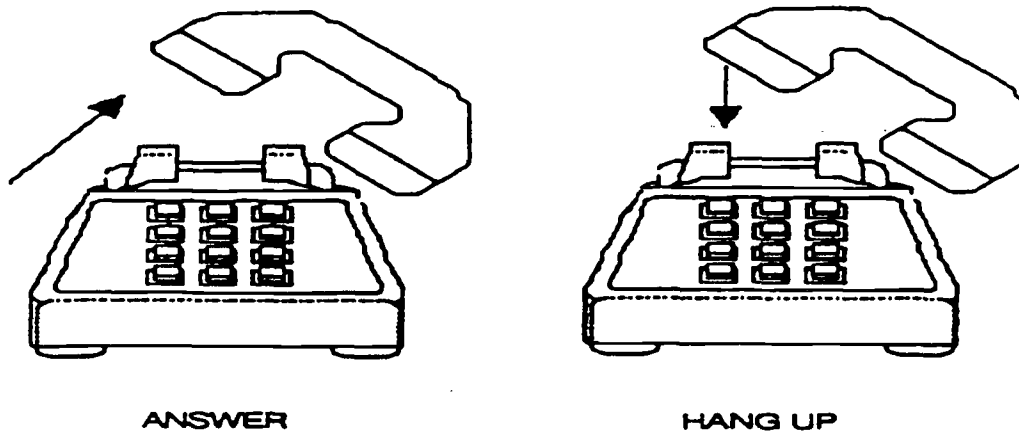


FIGURE 7.7: Example of Mirrored Icons

7.1.3.10 Icon Size

- a. Icons should be at least 1/4 inch in height to reduce the time required for positioning the cursor on the target and performing the required controlling actions.
- b. Up to five sizes of icon can be used for coding, but no more than three are recommended for tactical systems.
- c. When size coding of icons is used by operational systems, the larger icon should be 1.5 times as large as the next smallest.
- d. Icons should have a size ratio to the background of 1:1.5.

7.2 COMMAND ICONS

This section on "Command Icons" addresses the design and use of computer icons representing frequently used computer commands and operations. The concept of a universal set of icons will be discussed. The use of icons for "macro" instructions will be reviewed.

7.2.1 COMMAND ICON DESIGN

The following paragraphs provide guidelines for designing command icons. The designer is reminded that more research needs to be performed to arrive at a complete set of principles for the design of command icons. Icon design should be based upon a concrete, not abstract, concept with respect to the user. The designer should consider the display media when designing the icon (e.g., an icon design for a high-resolution display may lose meaning when presented on a low-resolution display).

7.2.1.1 Metaphor Selection

- a. The metaphor selected for command icon design should model the system being controlled.
- b. The metaphor selected for command icon design should be appropriate for the user's tasks, functions, and environment.
- c. The office metaphor may not necessarily be appropriate for tactical military applications.

7.2.1.2 Universal Icons

- a. To the extent possible, command icons should be standardized and consistent across all DoD applications.
- b. A common set of icons for command and utility functions may be possible within military applications. Research is needed to identify the metaphor and icon shapes that best represent the expectations of this user community.

7.2.1.3 Application-Specific Icons

- a. Command icon meaning should be consistent across displays and standardized within an application.
- b. COTS software must meet the consistency requirements for icon use and design within an application.
- c. COTS applications may use metaphor/icon designs that are not consistent with the suggested universal icon approach.

7.2.1.4 Recognition

- a. It is imperative to test icon design with a group of users who represent the intended user. This should ensure that the meaning of the icon is implicitly understood.
- b. The designer should create icons that have intrinsic meaning to the user. The user will associate a meaning with an icon. The stronger the associated meaning, the more easily the icon will be recognized and remembered.

7.2.1.5 User Expectation

Design command icons to provide a visual representation that matches the user's expectations and allows an association between the icon and the function being controlled.

7.2.2 COMMAND ICON APPLICATIONS

7.2.2.1 Usability

- a. Command icons should be general enough to allow the user to understand and use other metaphors or media, such as text-based systems.
- b. Application software should allow the creation of icons that represent "macro" instructions. The user will be able to use these macros more effectively with the advantages of a visual representation.

7.2.2.2 Existing Designs

- a. The designer should use existing icons that the user will recognize.
- b. Where possible, do not use icons specific to an application.

7.2.2.3 Unusual Designs

Avoid using unusual designs for icons.

REFERENCE LIST

<u>Paragraph</u>	<u>References</u>
7.0	Helander (1988); Gittins (1986); Lewis and Fallesen (1989)
7.1	Smith and Mosier (1986) para 3.1.8-5
7.1.1.1	Ziegler and Fähnrich (1988) p. 127
7.1.1.2	Ziegler and Fähnrich (1988) p. 127
7.1.1.3	Ziegler and Fähnrich (1988) p. 128
7.1.2	Ziegler and Fähnrich (1988) p. 128
7.1.3	Ziegler and Fähnrich (1988) p. 129
7.1.4	Bowser (1991)
7.2	Lewis and Fallesen (1989) p. 91
7.2.1	OSF (1990) p. 531; Lewis and Fallesen (1989) p. 92
7.2.2	Lewis and Fallesen (1989) p. 91
7.2.3	Gittins (1986) p. 528
7.2.4	OSF (1990) p. 531
7.2.5	Lewis and Fallesen (1989) p. 91; Gittins (1986) p. 540
7.2.6	Lewis and Fallesen (1989) p. 92; Gittins (1986) p. 530 and 540
7.2.7	OSF (1990) p. 530
7.2.8	Lewis and Fallesen (1989) p. 91; Gittins (1986) p. 540
7.3.1	Ziegler and Fähnrich (1988) p. 127
7.3.2.1	Smith and Mosier (1986) para 3.1.8-5
7.3.2.2	Gittins (1986) p. 526
7.3.3	Gittins (1986) p. 540
7.3.4	Ziegler and Fähnrich (1988) p. 128
7.3.5	Smith and Mosier (1986) para 2.4-13
7.3.6	Lewis and Fallesen (1989) p. 91; Gittins (1986) p. 539
7.3.7.1	Lewis and Fallesen (1989) p. 89
7.3.7.2	Lewis and Fallesen (1989) p. 89
7.3.7.3	Bullinger et al. (1987) p. 90; Gittins (1986) p. 536
7.3.8.1	Ziegler and Fähnrich (1988) p. 128
7.3.8.2	Lewis and Fallesen (1989) p. 91
7.3.9	Ziegler and Fähnrich (1988) p. 128; Smith and Mosier (1986) para 2.4-13; Shneiderman (1987) p. 200; Lewis and Fallesen (1989) p. 89; Lansdale (1988) p. 148
7.3.9.2	Ziegler and Fähnrich (1988) p. 128; Shneiderman (1987) p. 200; Lewis and Fallesen (1989) p. 89; Gittins (1986) p. 528

REFERENCE LIST (Cont.)

Paragraph

References

7.3.9.3	Lewis and Fallesen (1989) p. 91; Gittins (1986) p. 539
7.3.9.4	Lewis and Fallesen (1989) p. 91
7.3.9.5	Lewis and Fallesen (1989) p. 90; Gittins (1986) p. 539
7.3.9.6	Gittins (1986) p. 539
7.3.9.7	Gittins (1986) p. 539
7.3.9.8	Lewis and Fallesen (1989) p. 90
7.3.9.9	Lewis and Fallesen (1989) p. 92
7.3.10.1	Lewis and Fallesen (1989) p. 90
7.3.10.2	Lewis and Fallesen (1989) p. 90
7.3.10.3	Lewis and Fallesen (1989) p. 90
7.3.10.4	Lewis and Fallesen (1989) p. 90

8.0 COMMON FEATURES

The design of the user interface should consider tasks performed by the user as the primary criteria. Ease of programming software is a secondary consideration to factors that improve the application user interface.

This section describes features, functions, and field display formats that should be handled consistently by all DoD applications. Section 8.1 deals with issues that apply primarily to tactical systems. Section 8.2 and 8.3 apply to all DoD systems.

Section 8.2 discusses the topic of HELP. The purpose of HELP is to provide on-line assistance at the user's request. HELP information is not meant to tutor users but to assist them in recalling how to use an application. OSF/Motif and Open Look style guides define several types of HELP, and they should be consulted for details on how to implement the HELP features.

DoD systems appear to concentrate on using menu, function key, graphical interaction, and form-filling dialogs. Therefore, Sections 8.3 and 8.4 of this *Style Guide* document will deal with these techniques only. Information on query and natural language is presented in Section 11. Additional information on interactive dialog can be found in documents such as Smith and Mosier (1986), Helander (1988), or DoD (1989b).

8.1 TACTICAL SYSTEM COMMON FEATURES

This section describes features, functions, and field display formats that should be handled consistently by all DoD tactical applications.

8.1.1 DATE/TIME DISPLAY

When date and time information are displayed in digital form, the format should be as follows:

8.1.1.1 Date

Use YYMMDD, where YY is the last two digits of the year, MMM is the month, and DD is the date (e.g., 910104 specifies 4 January 1991), or DD

MMM YY, where DD is the day, MMM is the month, and YY is the year (e.g., 04 JAN 91).

8.1.1.2 Time

Use HHMM{SS}Z, where HH is the hour of a 24-hour day, MM is the minute, SS (optional) is the second, and Z is the time zone. Zulu (Z), or Greenwich Mean time in civilian terms, is the system standard and the default DoD display standard (e.g., 113024Z). Use colons or spaces on the display or output format to make the format more readable (e.g., 11:30:24Z). To simplify data entry and avoid extraneous characters, the colons or spaces should be generated as part of the form and not left to the user's discretion.

8.1.1.3 Local Time

Allow users to specify local time on hardcopy output and softcopy display, as desired (e.g., 11:30:24L). However, do not provide this option to users in operational systems where input and coordination are based on Zulu time.

8.1.1.4 Date/Time Group

Date/Time Group should be displayed as DDHHMMZ MMM YY, where DD is the day, HH is the hour of a 24-hour day, MM is the minute, Z is the time zone (defaults to Zulu), MMM is the month, and YY is the year (e.g., 041130Z JAN 91).

8.1.2 LATITUDE/LONGITUDE DISPLAY

Latitude and longitude displays will always be presented as two fields. The labels may be given as Lat and Long. The formats are as follows:

8.1.2.1 Latitude

Use D{D}H, where D (one or two characters) is the degrees of latitude and H is the hemisphere (N for North, S for South), or DD(MM{SS})H, where DD is the degrees of latitude, MM is the minutes of latitude, SS is the seconds of latitude, and H is the hemisphere (N for North, S for South).

8.1.2.2 Longitude

Use D(D{D})H where D (one, two, or three characters) is the degrees of longitude and H is the hemisphere (E for East, W for West), or DDD(MM{SS})H where DDD is the degrees of longitude, MM is the minutes of longitude, SS is the seconds of longitude, and H is the hemisphere (E for East, W for West).

8.1.3 USER-DEFINABLE PARAMETERS

All users should be able to configure their computer screens to meet individual preferences. User-definable parameters include, but are not limited to, the following:

8.1.3.1 Display Colors

As a minimum, users should be able to select map and window background colors from a color palette within a user-parameter selection window. The selected color should be immediately reflected in a sample item displayed within the selection window. However, users should not be allowed to change security banner colors or colors with specific tactical coded meaning. Other restrictions are noted in the service-specific addenda to this *Style Guide*.

8.1.3.2 Printer Default

In networked environments, users should be able to specify the printer destination.

8.1.3.3 Mouse Button Function Mappings

Users should be able to specify either left-handed or right-handed button configurations as defined in Section 3 of the *Style Guide*.

8.1.3.4 Help Level

Experienced users should be able to bypass novice level help messages that are beneficial to the new user.

8.1.4 WILD-CARD CHARACTERS

Wild-card characters are used in queries and searches to support patterns. The use of wild cards is application-specific. Some applications may disallow wild cards or restrict their use to only a few of the following wild card conventions. The following conventions should be used where possible:

8.1.4.1 Single Alphabetic Character

The use of an @ will replace any single alphabetic character (a-z and A-Z). For example, an input of abc@d would match abcad, abced, and abczd, but would not match abc7d or abcd4d.

8.1.4.2 Single Numeric Character

The use of a # will replace any single numeric character (0-9). For example, an input of 123#4 would match 12334, 12394, but would not match 123x4 or 123554.

8.1.4.3 Single Alphanumeric Character

The use of a ? will replace any single alphanumeric character (a-z, A-Z, 0-9, and punctuation marks). For example, an input of abc?d would match the character strings abcad, abc(d, abc'd, and abc7d, but would not match abcxxd.

8.1.4.4 String

The use of an * will replace zero or more alphanumeric characters. For example, an input of abc*d would match the character strings abcad, abcd, abckjfi(rjk)fid, and abc7d, but would not match abcd5.

8.2 ON-LINE HELP

On-line help (HELP) provides procedural aids, the ability to recover from errors, and advice without requiring the user to exit the application. Ideally, HELP is always available. A well designed system offers context-sensitive HELP.

Two elements are critical to HELP: the user interface and the content; both are equally important (Kearsley 1988). HELP should be easy to use and provide readily understandable user guidance. HELP must not add problems or make

the user's situation more confusing. The HELP interface design will contribute to how often HELP is used because the more difficult the interface is to use or access, the higher the probability HELP will not be used. No help application will be useful if it is difficult to obtain, hard to use, or not easy to return to the application program.

Computer users want to accomplish a particular task quickly and with the least effort possible. When users encounter a problem, they want a solution that involves minimal interruption of the task at hand. If information is not immediately available, users often guess, repeat a previous sequence, or ignore what is not understood. These responses usually lead to further problems.

Along with individual differences, users have varying degrees of computer experience. The following three types of user group may require different types or levels of HELP.

- Novices (users who have little experience with computers) will need help with basic concepts and operations. Novices usually want to see only necessary information.
- Experts (experienced computer users) want to know about limitations, shortcuts, complex operations, and anything else that will allow them to do their work more efficiently.
- Casual users (who may be either novices or experts) who only occasionally use a computer or current application, so they need help remembering aspects of the application they previously learned.

General guidelines:

- Make HELP easy for users to access
- Make HELP available throughout the application
- Make access to HELP uniform

- Make HELP easy to understand
- Make it easy to return to the application

8.2.1 TYPES OF HELP

HELP should reflect the user's requirements without significant impact on application response time. Of the following three types of HELP, the advice and active forms are preferred. Embedded training is often included in HELP. However, the recommendation is that embedded training not be combined with HELP.

8.2.1.1 Advice

When users query HELP, they find an interactive, context-sensitive source of information that indicates what entry to make at the current location in the application, the required keystroke, or which steps to be taken to complete the task.

8.2.1.2 Active

When the HELP application software senses an inappropriate entry, it interrupts to ask users what they are attempting and if they are sure they want to complete the operation they initiated. HELP then suggests the correct form or keystroke.

8.2.1.3 Passive

Users query HELP when they need assistance. The information may be in the form of on-line system documentation, such as a user's guide or a list of functions performed by combinations of keypresses.

8.2.2 GENERAL DESIGN

8.2.2.1 Minimize Keystrokes

Provide single keystroke access to and exit from HELP.

8.2.2.2 Provide Memory Aids

Assume users cannot remember everything required to run the application; provide memory aids.

8.2.2.3 Include Basic Information

Include basic information you would expect only novices to seek.

8.2.2.4 Expand Upon the Manual

Provide clearer explanations of information in the manual, using subsequent screens as needed. Do not simply repeat phrases from the manual the user has read but may not understand.

8.2.2.5 Choose On-line Portions of the Manual Selectively

Be selective when placing information from the user's manual on line; do not place the entire manual on line. This would be more difficult to navigate through and read than the hard-copy version and would waste system memory.

8.2.2.6 Include Obvious Information

Include all pertinent information, even that which may appear obvious to the developer.

8.2.2.7 Avoid Jargon

Avoid using jargon. A friendly and effective interface is the most important component of a HELP system. It is frustrating to a naive computer user to type "HELP," then receive a bit of cryptic jargon in reply. Ensure it is the jargon of all the users and not of the designer or programmer where use of jargon is unavoidable.

8.2.2.8 Do Not Overload the User

Do not expect the user to read about more than three HELP displays at a time or to remember more than about five points.

8.2.2.9 Do Not Use HELP to Teach

Do not use HELP to teach novices how to operate the system. Provide step-by-step instructions to remind occasional users how to perform the most common tasks. Remember that most users perform the same few tasks over and over, in the simplest possible way.

8.2.3 ACCESSIBILITY OF HELP

8.2.3.1 Universal Access

Provide access to HELP from every screen.

8.2.3.2 Availability

Remind users that HELP is easily available by displaying the command or function key used to get HELP.

8.2.3.3 Display HELP Status

Display a message indicating the status of HELP availability if HELP is not available at all times or places in the program.

8.2.3.4 Single Action to Invoke

Make it possible for users to get HELP using only a single keypress or mouse-click.

8.2.4 PROVIDE HELP ON HELP

8.2.4.1 Alphabetical Index of Functions

Make an alphabetical index of HELP functions available to the user.

8.2.4.2 Alphabetical Index of Commands

Provide an alphabetical index with explanations of all commands used by the application software, showing the argument options.

8.2.4.3 Show How to Use

Show users how to use the HELP function. Never assume it is obvious, even to expert users.

8.2.4.4 Present Alternatives

Show how to get HELP from anywhere in the system. Users may know only one route, so detail the alternatives, including how quick and easy it is to use the options. Define the different meanings of the HELP display and explain their functions.

8.2.4.5 Navigating Through HELP

Show how to navigate within HELP. Explain how to scroll or page through a topic and how to jump to related topics.

8.2.4.6 Provide HELP on Screens and Windows

Describe the current window, including its function and tasks the user can perform.

8.2.4.7 Provide Instructions to Novices

To assist novice and casual users, put instructions for using HELP on every HELP display.

8.2.4.8 Instruct on When to Use

Provide users with complete instructions on when to use the information supplied by HELP.

8.2.5 APPLICATION INFORMATION

Provide a list of application capabilities. Show the application components, options, and structure to help the user understand the application and use it more effectively. Experienced users as well as novices underutilize many applications because they do not recognize the full range of capabilities.

8.2.5.1 Provide Shortcuts

Use HELP to point out shortcuts and unused features.

8.2.5.2 HELP on Error Messages

Make available successively more detailed explanations of a displayed error message.

8.2.5.3 HELP on Prompts and Definitions

Make available successively more detailed explanations of a displayed question or prompt and definitions of specified terms.

8.2.5.4 Show Correct Input

Provide examples of correct input or valid commands.

8.2.5.5 Show Command Format

Provide a description of the format of a specified command and a list of allowable commands.

8.2.5.6 Provide User-Centered HELP

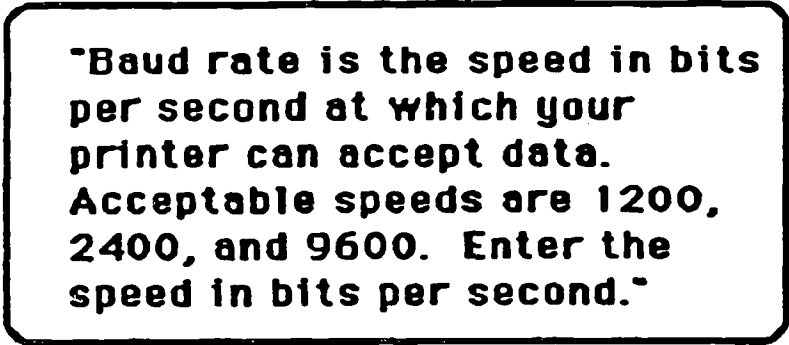
Make the HELP user-centered; base it on the user's task, not on application characteristics. Descriptions of application characteristics are more appropriate for a hard-copy user's manual.

8.2.6 PROVIDE HELP IN CONTEXT

Context-sensitive HELP may be the most important kind of HELP for users. Context-sensitive HELP should describe the nature of a specific control (check button, radio button, slider bar) and how people use that control.

8.2.6.1 Provide Specific HELP

Ensure the HELP is specific to each level of user interaction (e.g., for context-sensitive HELP in a field specifying printer baud rate). See Figure 8.1.



"Baud rate is the speed in bits per second at which your printer can accept data. Acceptable speeds are 1200, 2400, and 9600. Enter the speed in bits per second."

Figure 8.1: Printer Baud Rate

8.2.6.2 Show Correct Alternatives

List correct alternatives if the user enters an incorrect command.

8.2.6.3 Provide HELP Within Application

Provide HELP within the application so users do not have to abandon their place in the application to seek HELP. Users should not have to close files, exit the application, and/or log off to invoke a HELP utility.

8.2.6.4 Use Split Screen or Window

Allow users to see the application screen that relates to the HELP request by means of a split screen or window. A separate HELP screen that completely replaces the application screen is undesirable because it prevents the user from simultaneously observing the problem and the HELP screen.

8.2.6.5 Resize and Reposition HELP Windows

Provide the user the capability to resize and reposition windows to see the HELP information and the problem at the same time.

8.2.6.6 Identify Special Keys

Display the meanings assigned by the application where applications have special uses for keys, especially function keys.

8.2.7 USER CONTROL OF THE HELP SYSTEM

The more control users have over the help system, the more useful they will find it.

8.2.7.1 User-Initiated

Allow users to initiate a HELP request and select the desired HELP topic.

8.2.7.2 User-Selected Levels

Allow users to select a level of HELP if multiple levels are available.

8.2.7.3 Annotate Messages

Allow users to annotate existing HELP messages.

8.2.7.4 Describe Key Functions

Provide the capability within HELP of pressing any key to obtain a list of features whose names begin with that letter. When the user selects a feature from the list by highlighting or clicking, provide an explanation of the feature.

8.2.8 PROVIDE CONSISTENT HELP FORMAT

8.2.8.1 Consistent Screens

Provide consistent HELP aids from screen to screen, both with indicators that HELP is available (e.g., "F1 = HELP") and the specific location on the screen.

8.2.8.2 Progressive Detail

When providing progressively more detailed explanations, ensure the process of moving from level to level is consistent from screen to screen.

8.2.9 SELF-EXPLANATORY AND CONCISE DISPLAYS

8.2.9.1 Match Titles to Contents

The content of a HELP window should be reflected in its title (e.g., the title of a HELP window for the entry field "Trans" could be "Help for Trans").

8.2.9.2 Match Names

Ensure the name on the HELP display matches the panel from which help was requested (e.g., when working on an accident report, the help display may read, "HELP: ACCIDENT REPORT").

8.2.9.3 Ensure Relevancy to the User

Tailor the display to the current information requirements of the user, so only relevant data are displayed.

8.2.9.4 Provide Clear Messages

Make error and HELP messages clear, concise, and appropriate to the experience and training users have had in using the system.

8.2.9.5 Use Task-Oriented Wording

Adopt task-oriented wording for labels, prompts, and user guidance messages, incorporating whatever special terms and technical jargon may be normally employed in the user's tasks.

8.2.9.6 Increase Understandability

To increase understandability of HELP, apply the following principles:

- a. Use short sentences when writing HELP messages.
- b. Use the active voice in all HELP messages.
- c. Provide as many examples as possible for each HELP screen.
- d. Place HELP information in tables, where applicable.
- e. Put the answer before the explanation when presenting HELP information in the

dialog.

- f. Answer the most likely HELP questions immediately.
- g. Minimize the user requirement to scroll or page through displays.

8.2.10 MAKE RETURN TO APPLICATION EASY

8.2.10.1 Single Keystroke

The user should be able to return to the application with only a single keypress or mouse-click.

8.2.10.2 Exit HELP Easily

When a single keystroke exit is not possible. The user should be able to return to the application easily, without calling up a menu, then choosing an item from it.

8.2.11 KEEP HELP CURRENT

8.2.11.1 Provide Up-to-Date HELP

Plan and build HELP concurrently with developing applications so HELP information reflects the current version of the software. Provide updates to HELP with subsequent software releases.

8.2.11.2 Tailor HELP to the User

Collect data on user target population to tailor HELP to the training and experience of the users.

8.2.12 PROVIDE USER OPTIONS

8.2.12.1 Bookmarking

Provide "bookmarking" so users can tag specific HELP messages for easy referral later. This can be especially useful in a large help system consisting of many topics and screens. Bookmarking allows users to customize the help system to their own needs and filter out information of no interest; it can speed up the HELP process and return the user to work with fewer interruptions.

- a. Allow the user to select a bookmark option while viewing the message to tag a HELP message.
- b. Ensure the user has an option to see all or just the bookmarked messages.
- c. Ensure a print option is available while HELP messages are being displayed. Users often want to print out HELP information to study it further.

8.2.13 SYSTEM-INITIATED MESSAGES

Provide system-initiated messages when an error has been detected or when there is other evidence the user is having a problem (e.g., missing parameters, duplicating erroneous commands, long lapses in response, out-of-range responses, etc.).

8.2.13.1 Positive Tone

Messages should have a positive tone, indicating what must be corrected. Focus on correcting the problem, not the action that caused it.

8.2.13.2 User Control

Present system-initiated messages to users as advice or suggestions. HELP messages should not be intrusive.

8.2.13.3 Error Control

Provide system-initiated HELP messages for systems where incorrect user actions could result in serious consequences. This is especially true for destructive actions such as deletions (e.g., MS-DOS command: "Del *.*"), file replacements, exiting an application without saving data, or renaming a file.

8.2.13.4 Document Errors

Error messages should always state the error detected, the input field containing the error, and the corrective action.

8.2.13.5 User Understanding of Message

Messages should clearly indicate whether they are meant to inform of error, indicate status, prompt for action, or provide feedback.

8.2.13.6 User Options

Give users the option to turn off system-generated messages or to specify the level or type of message to be given (e.g., advisory, caution, warning).

8.2.13.7 Avoid Jargon

Error messages should be specific and address the problem in user terms, avoiding vague terms such as "syntax error" or obscure error code numbers.

8.3 INTERACTIVE CONTROL

Interaction between the computer and the user is performed through a two-way communication process: 1) the user inputs commands, and 2) the computer responds to the input. Generally, two interchangeable names are given to this process - sequence control (Smith and Mosier 1986) and interactive control (DoD 1989a). The term "interactive control" will be used in this *Style Guide*.

Interactive control of a system occurs through a give-and-take of command and response between the user and the computer, called a "dialog." The following have been identified as the eight major types of user-computer dialogs (Smith and Mosier 1986):

- Question and Answer - The user responds to questions posed by the computer.
- Form Filling - The user enters a series of commands or data items in predefined fields. These fields may be mandatory or optional.
- Menu Selection - The user selects from predefined option lists by pointing with a device, such as a mouse, or keying in associated codes.
- Function Keys - The user controls the dialog by using fixed or variable function keys on the keyboard.
- Command Language - The user makes control entries by composing specified messages for the computer.
- Query Language - The user employs a specialized type of command language to elicit information from a computer system. This is used extensively with databases.

- Natural Language - The user can compose messages to control the computer based on natural, not specialized, languages.
- Graphical Interaction - The user makes selections and controls the computer interaction by direct manipulation.

Each of these eight types of dialogs can be used individually or combined as a suite or set of techniques. Open Software Foundation (OSF)/Motif, for example, supports a combination of menu and graphical interaction techniques.

Eight principles form the basis for designing a good human-computer dialog (Shneiderman 1987; Bailey 1982). These principles are as follows:

- Strive for consistency of design across terminology, menus, command structure, etc. for all applications.
- Enable frequent users to use shortcuts, improving user acceptance and overall system performance.
- Offer informative feedback for all user actions.
- Design dialogs to yield closure. The user will then feel a sense of accomplishment and will know when to go on to the next task.
- Offer simple error handling, both by system error-checking and ease in correcting an identified error.
- Allow easy reversal of actions, such as an UNDO capability.
- Enable the user to feel in control of the interaction with the system.
- Reduce the short-term memory load on the user by means of intuitive displays and interactive sequences, sufficient training, and on-line helps and tutorials.

The primary dialog types used by applications share certain design considerations and guidelines. These are addressed in this section and organized into six topics: general, context definition, transaction selection, interrupts, error management, and alarms.

8.3.1 GENERAL

The following general guidelines for interactive control apply to DoD systems.

8.3.1.1 Displayed Context

If the results of a control entry vary depending on a prior action of the user or computer, display a continuous indication of the current context (mode).

8.3.1.2 Irrelevant Data

The user should have the capability to remove irrelevant items from the display and to reverse this action (i.e., retrieve information that was removed).

8.3.1.3 Page-Back Capability

When the requested data exceed the capacity of a single display frame, give the user some easy methods of moving back and forth over displayed material by paging or panning/scrolling.

8.3.1.4 Upper and Lower Case Equivalent

For interpreting user-composed control entries, treat upper and lower case letters as equivalent.

8.3.1.5 User-Callable Unfamiliar Term Descriptions

- a. Interface dialog should be written to provide the capability for the user to call up descriptions of unfamiliar terms and commands through context-sensitive HELP screens. See Section 8.2 for additional information.
- b. The interface should display to the user, as needed and in immediately usable form, the terminology and commands necessary to perform the task associated with the displayed information.

8.3.1.6 User-Paced Sequence Control

Allow users to pace control entries, rather than forcing them to keep pace with computer processing or external events.

8.3.1.7 Logical Transaction Sequences

Design the sequence of transactions (e.g., number and sequence of steps in a

task) from the perspective of what is logical to the user, not what is logical from the perspective of computer processing or ease of programming.

8.3.1.8 Automated Information Entry

Routinely and automatically include informational elements required for every communication or transaction after first input (e.g., call signs and authentication procedures).

8.3.1.9 Customized Display/Control Options

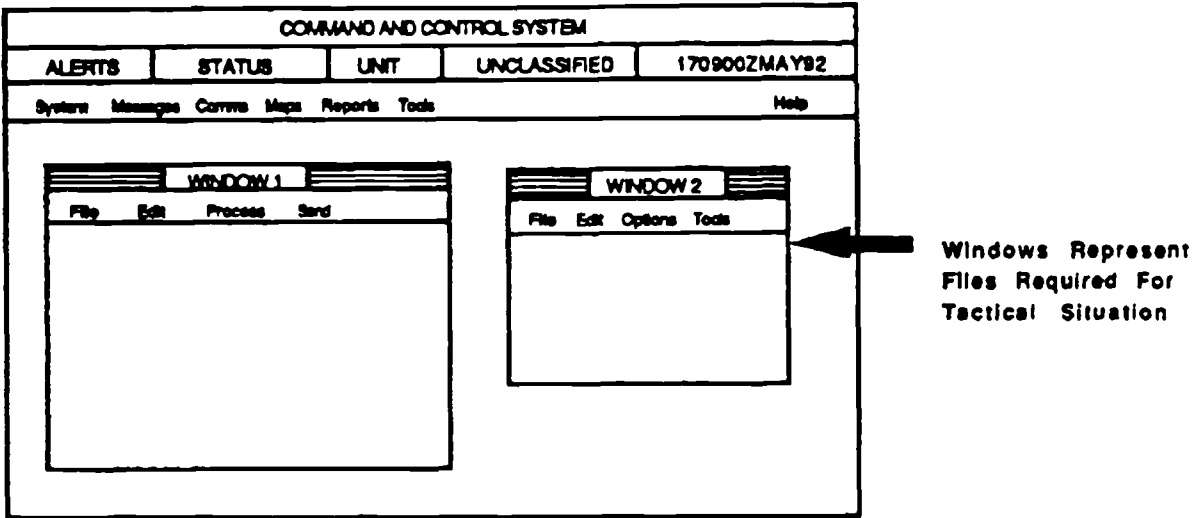
Allow the user to customize the information displayed on a screen to the particular tactical mission or scenario. For example, the user should have the flexibility to define which files can be displayed concurrently and what tactical data will be utilized in a single display. See Figure 8.2 for examples.

8.3.1.10 Distinctive Display of Control Information

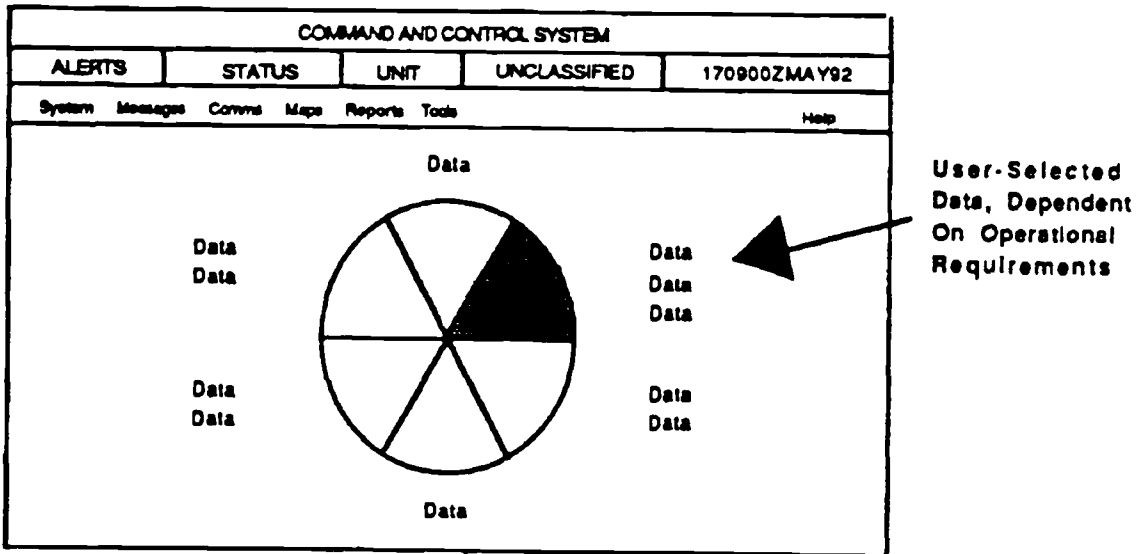
Design all displays so features (e.g., prompts, advisories, etc.) relevant to interactive control are distinctive in position and/or format.

8.3.1.11 System Matched to User Abilities

Ensure applications are adaptable to individual differences and accommodate the variety of user abilities, whether novice or expert. For example, accelerator keys for menu selection or command stacking should be available to the expert.



To Display Required Files



To Display Specified Information

FIGURE 8.2: Example of How a Screen Display Can Be Customized

8.3.1.12 Response Demand

Demand response from the user while instructions on how to respond are still visible on the display.

8.3.1.13 Consistency

- a. Ensure interactive control actions are consistent in form and consequence. Employ similar means to achieve similar ends, from one transaction to the next and from one task to another throughout the command and control system.
- b. Ensure the results of any control entry are compatible with user expectations, so a change in the state or value of a controlled element is displayed in an expected or natural form. For example, NEXT PAGE should call up the next page of the active file, not of an unrelated file.
- c. When selecting names for interactive control functions, choose names that are semantically congruent with natural usage, especially for paired opposites. For example, to move a cursor up, use UP. For the opposite command, use DOWN, not LOWER.

8.3.1.14 Control

- a. Allow the user to complete a control entry or action through an explicit action, such as ENTER, before the system interrupts to indicate a computer-recognized word.
- b. Ensure control actions are simple, particularly for real-time tasks such as fire control that require rapid user response. Control logic should permit completion of a task with the minimum number of actions, consistent with user abilities.
- c. Allow the user to make control entries as needed, in essence, stacking commands.
- d. Ensure that the sequence of control entries is not slowed by delays in computer response. In general, system response time should be in the range of 5-50 milliseconds and no longer than 0.2 seconds. System response time, in this context, refers to the time between keystroke and screen response. It does not refer to response time for a query of a database.

8.3.1.15 Feedback

- a. Ensure the system provides periodic feedback that indicates normal operation is occurring if the user waits more than 15 seconds for the computer to respond.
- b. Ensure that the computer acknowledges every control entry immediately; for every action by the user, some reaction from the system should be apparent.
- c. When computer processing in response to a control entry is lengthy, give the user overt and positive indication of when the processing has been completed.
- d. Provide displayed feedback for all user actions; display keyed entries stroke by stroke.

8.3.1.16 Lockout

- a. If application processing time requires a delay of concurrent user inputs and no keyboard buffer is available, the keyboard should be locked out until the computer is ready to accept the next input.
- b. When keyboard lockout has been terminated, provide a clear indication to the user.
- c. In situations where control lockout occurs, provide the user with a means of aborting the transaction that caused the lockout. A method such as a special function key can accomplish this transaction. The system should not reset and lose previous processing when aborted. The system should provide the option of resetting the system.

8.3.1.17 Response Time

- a. Ensure that the speed of computer response to user entries is appropriate to the type of dialog. Also ensure that responses to menu selections, function keys, and most entries during graphic interaction are immediate.
- b. Ensure that the speed of computer response to user control entries is appropriate to the transaction involved. Generally, those transactions perceived by a user to be simple should have faster responses.

8.3.1.18 Cursor Design

- a. Indicate the current cursor position by displaying some distinctive cursor symbol at that point. In all cases, try to obtain the highest contrast possible between the cursor and the background. Cursor size should be such that the cursor is not lost in the "clutter" of the background. A contrast ratio of 3:1 is the minimum recommended for an office environment.
- b. Provide the user with an easy, accurate means of pointing a displayed cursor at different display elements and/or display locations. The cursor positioning should work consistently throughout the application.
- c. For most graphics data entry, pointing should be a dual action, first positioning a cursor at a desired position, then confirming that position to the computer.

8.3.2 CONTEXT DEFINITION

8.3.2.1 Application-Provided Context Definition to the User

Design the interactive control of the application such that the user maintains an understanding of the context for the task being performed. Ensure the system prompts expected user actions. For example, display the results of previous steps in the task affecting the present step and current options.

8.3.2.2 Context Established by Prior Entries

Design the interactive control software to interpret current control actions in the context of previous entries; do not require the user to re-enter data. Prompt the next logical action by the user.

8.3.2.3 Record of Prior Entries

Allow the user to request a summary of the results of prior entries (i.e., a history file) to help determine present status.

8.3.2.4 Display Operational Mode

When context for a user task is defined by the operational mode, display to the user the current mode and any other pertinent information.

8.3.2.5 Consistent Display of Context Information

Ensure information displayed to provide context for interactive control is distinctive in location and format and consistently displayed from one transaction to the next throughout all related applications.

8.3.2.6 Highlighting Selected Data

When a user is performing an operation on some selected display item, highlight that item.

8.3.2.7 Display Control Parameters

Allow the user to review any active control parameter(s).

8.3.3 TRANSACTION SELECTION

8.3.3.1 Consistent CONTINUE Option

At any step in a defined transaction sequence, if there is only a single appropriate next step, provide a consistent control option, such as ENTER, to continue to the next transaction.

8.3.3.2 Indicating Control Defaults

When control is accomplished by keyed command or option code entries, and a default is defined as a null control entry, indicate that default to the user. See Figure 8.3.

8.3.3.3 User-Specified Transaction Timing

When appropriate to task requirements, allow the user to specify transaction timing. For example, the user should be able to specify when a requested transaction should start, when the transaction should be completed, and/or the periodic scheduling of repeated transactions.

8.3.3.4 Display Option Codes

When the user must select options by code entry, display the code associated with each option in a consistent, distinctive manner.

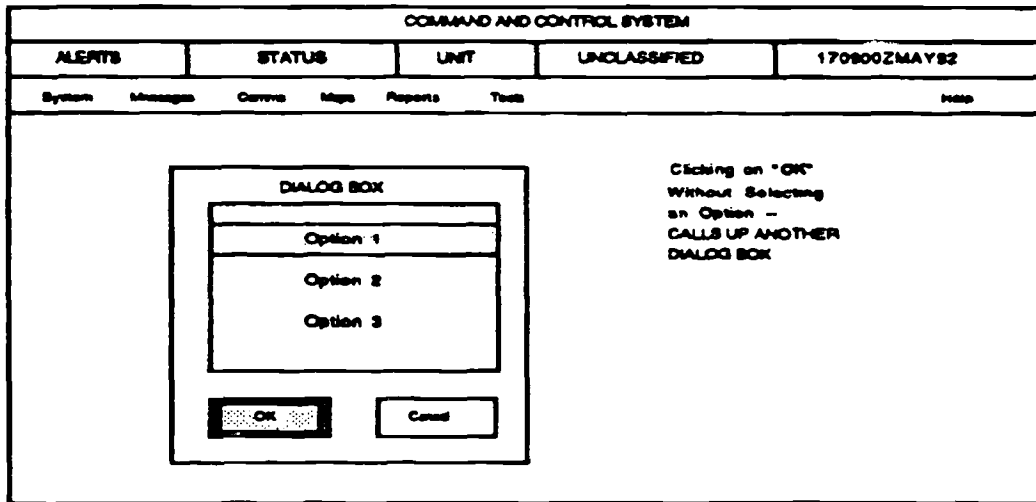


FIGURE 8.3: Example of How a System Can Display Default Status

8.3.3.5 Available Options

Ensure the user can clearly distinguish between available and unavailable options. Displaying unavailable options in a visually distinctive manner may aid navigation.

8.3.3.6 Stacked Control Entries

Allow the user to key a sequence of commands or option codes as a single stacked control entry. Stacked control entries, also called stacked commands, allow the user to input a series of command entries at one time. This can be done by continuous entry while the computer processes previous commands, or by typing in a series of commands and then entering them simultaneously.

- a. For control entry stacking, accept command names or their abbreviations or option codes, just as if those control entries had been made separately.
- b. Provide flexibility in transaction selection by allowing the user to assign a single name to a defined series of control entries. Use that named macro for subsequent command entry. All applications should include a predefined informational query process (see Section 11).

- c. For control entry stacking, require entries be made in the order they would normally be made when performing a succession of separate control entry actions.

8.3.3.7 Cursor Placement

- a. When the user must select options by keyed entry of a corresponding code, place the cursor in the control entry area at display generation.
- b. When the user will need to select among displayed options by pointing, place the cursor on the first (most likely) option at display generation.

8.3.3.8 Prompting Control Entries

Provide the user with whatever information may be needed to guide control entries. Incorporate prompts in a display at any point in a transaction sequence, and/or provide prompts in response to requests for HELP.

8.3.3.9 General List of Control Options

Provide a general list of basic control options that will always be available to serve as a home base or consistent starting point for control entries. For an example, see Figure 8.4 below (see also Paragraph 8.2.3.3).

8.3.4 INTERRUPTS

8.3.4.1 REVIEW Option

If appropriate, provide a nondestructive REVIEW option that will return to the first display in a defined transaction sequence, permitting the user to review a sequence of entries and make necessary changes.

8.3.4.2 PAUSE and CONTINUE Options

If appropriate, provide PAUSE and CONTINUE options that will interrupt and later resume a transaction sequence without any change to data entries or control logic for the interrupted transaction.

COMMAND AND CONTROL SYSTEM						
ALERTS	STATUS	UNIT	UNCLASSIFIED	170900ZMAY92		
System	Messages	Comms	Maps	Reports	Tools	Help

FIGURE 8.4: Example of a General List of Control Options

8.3.4.3 Indicating PAUSE Status

If a PAUSE option is provided, display some indication of the PAUSE status whenever that option is selected by a user, and prompt the CONTINUE action that will permit resumption of the interrupted transaction.

8.3.4.4 END Option

If appropriate, provide an END option that will conclude a repetitive transaction sequence.

8.3.4.5 Aborting or Escaping from a Function

Ensure that the system makes it easy for the user to abort, escape, or exit from a current operation or function (see Paragraph 8.3.5.9).

8.3.4.6 Indicating System Status

Inform the user that system action is continuing. Ensure that the "working" indicator has dynamic aspects to keep the user informed of continuing system function.

8.3.4.7 SUSPEND Option

- a. If appropriate, provide a SUSPEND option that will preserve current transaction status when a user leaves the system and permit resumption of work when the user later logs back onto the system.
- b. If a SUSPEND option is provided, display some indication of the SUSPEND status whenever a user selects that option. Prompt the user with those procedures that permit resumption of the suspended transaction at the subsequent log-on. For example, specifically prompt the user with "Type EXIT to return to application."

8.3.4.8 System Interruptions

Ensure that the system interrupts the user only when necessary to prompt response, to provide essential feedback, and to signal errors.

8.3.4.9 CANCEL Option

If appropriate, provide a CANCEL option that will erase changes just made by the user and restore the current display to its previous version.

8.3.4.10 Distinctive Interrupt Options

If different types of user interrupts are provided, design each interrupt function as a separate control option with a distinct name.

8.3.4.11 GOBACK Option

If appropriate, provide a nondestructive GOBACK option that will display the previous transaction.

8.3.4.12 RESTART Option

If appropriate, provide a RESTART option that will cancel entries made in a defined transaction sequence and will return to the beginning of the sequence. When data entries or changes will be nullified by restart, require the user to CONFIRM.

8.3.5 ERROR MANAGEMENT

8.3.5.1 User Confirmation of Destructive Entries

When a control entry (including log-off) will cause an extensive change in stored data, procedures, and/or system operation, and particularly if it cannot be easily reversed, notify the user and require confirmation of the action before implementation.

8.3.5.2 User Warned of Potential Data Loss

Word the prompt for a CONFIRM action to warn the user explicitly of any possible data loss.

8.3.5.3 Errors in Stacked Commands

If an error is detected in a stacked series of command entries, ensure that the system either consistently executes to the point of error or consistently requires the user to correct errors before executing any command.

8.3.5.4 Partial Execution of Stacked Commands

If only a portion of a stacked command can be executed, notify the user and provide appropriate guidance to permit correction, completion, or cancellation of the stacked command.

8.3.5.5 Flexible GOBACK for Error Correction

Allow the user to GOBACK easily to previous steps in a transaction sequence in order to correct an error or make any other desired change.

8.3.5.6 Explicit Entry of Corrections

When the user has completed correcting an error, whether a command entry or data entry, require an explicit action to re-enter the corrected material. Use the same ENTER action for re-entry that was used for the original entry.

8.3.5.7 Prompting Command Correction

If an element of a command entry is not recognized or is logically inappropriate, the system should prompt the user to correct that element, rather than require re-entry of the entire command.

8.3.5.8 Immediate Data Correction

If a data entry transaction has been completed and errors are detected, allow the user to make corrections directly and immediately.

8.3.5.9 Distinctive Confirm Action

Provide an explicitly labeled confirm control, such as a function key or widget (e.g., control button or dialog box) different from the ENTER control, for user confirmation of questionable or destructive control and data entries. See Figure 8.5.

8.3.5.10 UNDO to Reverse Control Actions

Ensure that any user action can be immediately reversed by an UNDO command.

8.3.5.11 Appropriate Response to All Entries

Design software to provide an appropriate response for all possible control entries, correct and incorrect. For example, selecting an incorrect function key should cause a message indicating the appropriate selections.

8.3.5.12 Appropriate Terms For All Entries

Ensure software is consistent in the use of terms, and use only the most explicit term. Use "cancel" for cancel functions, rather than a simple acknowledgement such as "OK." Avoid complex terms (i.e., "Save & Apply")

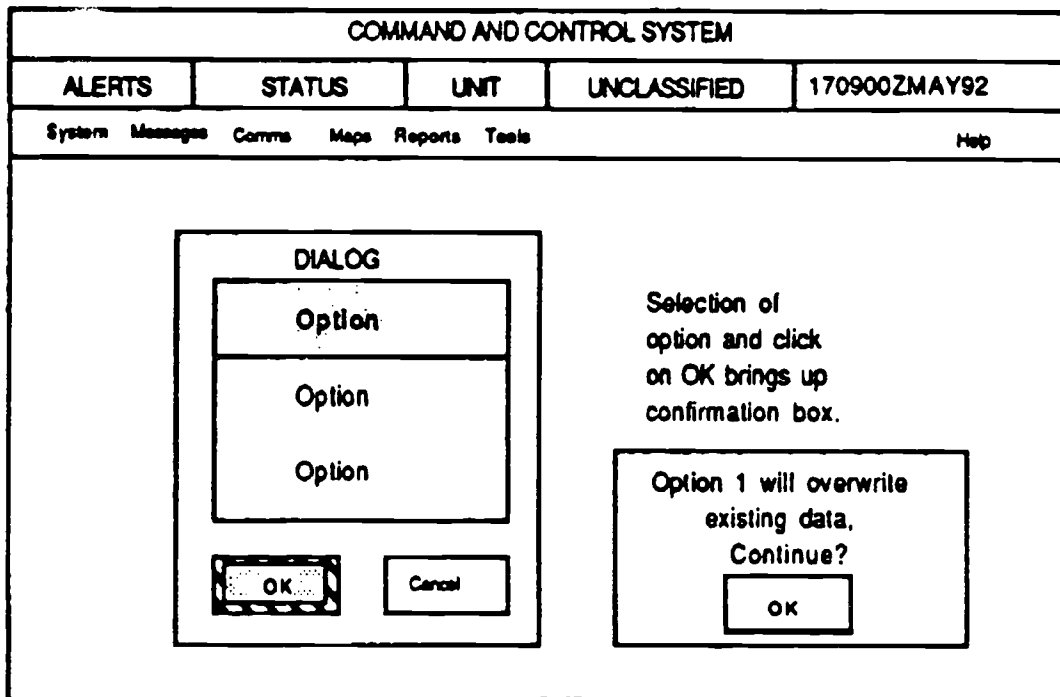


FIGURE 8.5: Example of a Distinctive Confirm Action, Using a Dialog Box

or "Exit to Prior Screen"), if possible. Ensure complex terms have one consistent meaning within an application.

8.3.5.13 Display Duration

Ensure notices, alerts, and informational displays remain visible to the user until responded to by specific user action. Field use of computers creates a situation where the user may not be continuously monitoring the screen presentation. Therefore, automatic time-outs should not be used where mission critical information is displayed.

8.3.5.14 Selection Errors

The pointing device interface uses both single and double clicks for control actions. The software needs to protect the system from inadvertent double clicks by the user. The protection supplied should be consistent with user and

system requirements.

8.3.5.15 Inappropriate Item Selection

The user should be cued, but not allowed to select items that are not available. Output fields should not allow data entry without the user acknowledging selection of the option. The software should prevent data entry in any field that is not appropriate.

8.3.6 ALARMS

8.3.6.1 Special Acknowledgment of Critical Alarms

When the user must acknowledge special or critical alarms in a unique way, such as a special combination of key strokes, ensure this acknowledgment does not inhibit or slow the response to the condition initiating the alarm.

8.3.6.2 Alarm Reset

Provide the user with a simple means of turning off an auditory alarm without erasing any displayed message that accompanies the auditory signal. For noncritical alarms, provide a simple method for acknowledging and turning off the signal.

8.3.6.3 Distinctive and Consistent Alarms

Ensure alarm signals and messages are distinctive for each class of event, such as INCOMING MESSAGE ALERT, TERMINAL STATUS, TRACK ALERT, etc.

8.3.6.4 Alarm Definition by User

When monitoring tactical situations or tactical data status, allow the user to define the conditions (such as priorities, percentages, target flight path, etc.) that result in a software-generated alarm, alert, or status message.

8.4 FUNCTION KEYS

The two types of function key are fixed and variable. The fixed key has only one predefined function associated with it. The variable key function will vary depending on the system mode or level within the interactive dialog. The

function for the variable key is communicated to the user by changing the label located adjacent or internal to the key or through soft keys. Soft keys are objects on the display screen that represent the function keys on the keyboard. As the function of a key changes, the soft key labeling also changes. Fixed and variable function keys can be used together and with other dialog methods.

As with any interactive control method, the designer should be aware of the following overarching design guidelines:

- consistent design in terms of placement, labeling, and procedural logic
- easy association with the function being called up through labeling located adjacent to the function keys
- feedback
- spatial consistency between the labeling and the function key

8.4.1 GENERAL

Function keys are located on the keyboard and activate a computer software function when pressed.

8.4.1.1 Usage

Function key dialog should be considered for:

- a. frequently required control entries
- b. tasks requiring only a limited number of control entries or in conjunction with other dialog types as a ready means of accomplishing critical entries that must be made quickly, without syntax error
- c. interim control entries (i.e., for control actions taken before the completion of a transaction)

8.4.1.2 Feedback for Function Key Activation

When function key activation does not result in any immediately observable response from the computer, provide users with some other form of computer

acknowledgment and feedback. No system function should be activated without an indication to the user.

8.4.1.3 Disabling Unneeded Function Keys

When function keys are not needed for any current transaction, temporarily disable those keys under computer control; do not require the user to apply mechanical overlays for this purpose (see Paragraph 8.4.1.7).

8.4.1.4 Function Key Meaning

In general, each function key should control only one function. If a key must control more than one function, display the actual or current meaning of the function to the user by displaying soft keys on the screen.

8.4.1.5 Soft Key Design

Soft function keys displayed on the screen should be located close to the actual keyboard function keys and in the same spatial orientation. For example, on command and control system keyboards with function keys across the top, place soft keys at the bottom of the screen, directly above the keyboard as illustrated

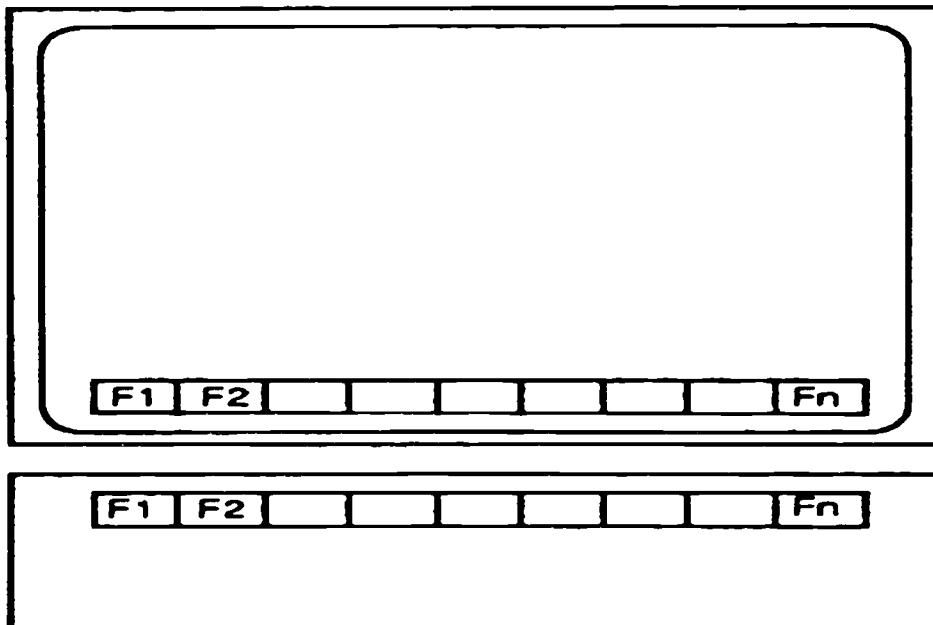


FIGURE 8.6: Example of Soft Key Location

illustrated in Figure 8.6.

8.4.1.6 Redundant Activation of Soft Key Function

The user should be able to activate the function represented on a soft key through either the function key or a pointing device, such as a mouse.

8.4.1.7 Indicating Active Function Keys

If some function keys are active and some are not, indicate the current subset

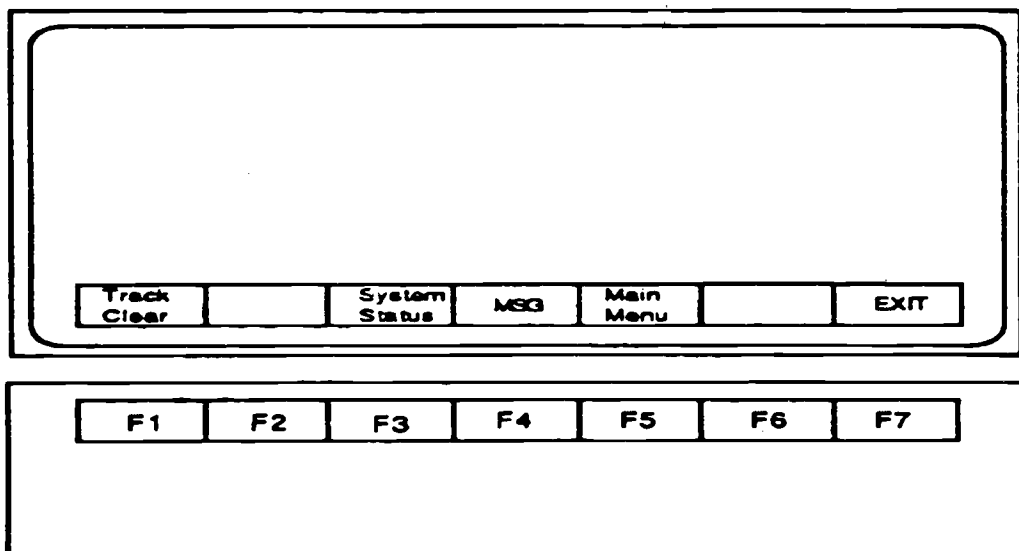


FIGURE 8.7: Suggested Method for Indicating Active and Inactive System Function Keys

of active keys in some noticeable way, such as brighter illumination or blanking of corresponding soft key labels on the display. See Figure 8.7.

8.4.1.8 Key Functionality Load

Avoid overloading the functionality of keys, however, provide the user with all necessary function controls required to perform the task.

8.4.1.9 Easy Return to Base-Level Functions

If functions assigned to a set of keys change as a result of user selection, give

the user an easy means to return to the initial, base-level functions or menu. See Figure 8.8.

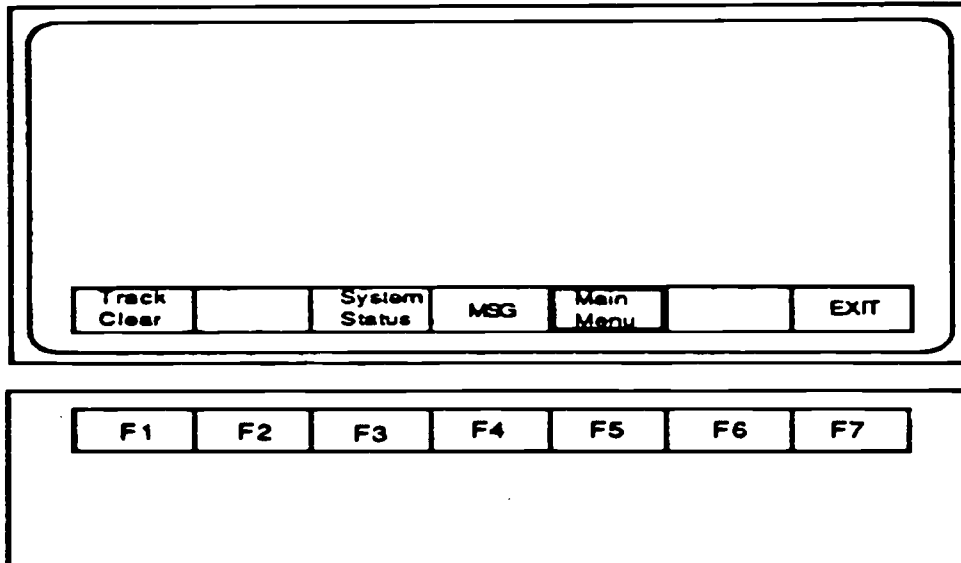


FIGURE 8.8: Recommended Method for a Return to Base-Level Functions

8.4.2 CONSISTENCY

8.4.2.1 Consistent Functions in Different Operational Modes

When a function key performs different functions in different operational modes, assign equivalent or similar functions to the same key.

8.4.2.2 Consistent Assignment of Function Keys

If a function is assigned to a particular key in one computer transaction, assign that function to the same key in other transactions.

8.4.3 DOUBLE KEYING

8.4.3.1 Logical Pairing of Double-Keyed Functions

If double (control/shift) keying is used, the functions paired on one key should be logically related to each other.

8.4.3.2 Consistent Logic for Double Keying

If double (control/shift) keying is used, the logical relation between shifted and unshifted functions should be consistent from one key to another.

8.4.4 LABELING

8.4.4.1 Distinctive Labeling of Function Keys

Label each function key informatively to designate the function it performs; make labels sufficiently different from one another to prevent user confusion.

8.4.4.2 Labeling Multifunction Keys

If a key is used for more than one function, always indicate to the user which function is currently available.

8.4.4.3 Labeling of Menu Options for Function Keys

When designing a command and control menu where options are selected through variable function keys, avoid using a function key number (e.g., F1, F2) as option designator. Instead, place the function key label just above the key on the display. See example in Figure 8.9.

8.4.5 LAYOUT

8.4.5.1 Layout Compatible with Use

Make layout of function keys compatible with their importance. Give keys for emergency functions a prominent position and distinctive coding (e.g., size and/or color).

8.4.5.2 Safeguards

Provide physical protection, software disabling, or interlocks for keys with potentially disruptive consequences.

8.4.5.3 Distinctive Location

Group function keys in distinctive locations on the keyboard to facilitate learning and use; place frequently used function keys in the most convenient

locations. For command and control systems, this should be at the top of the

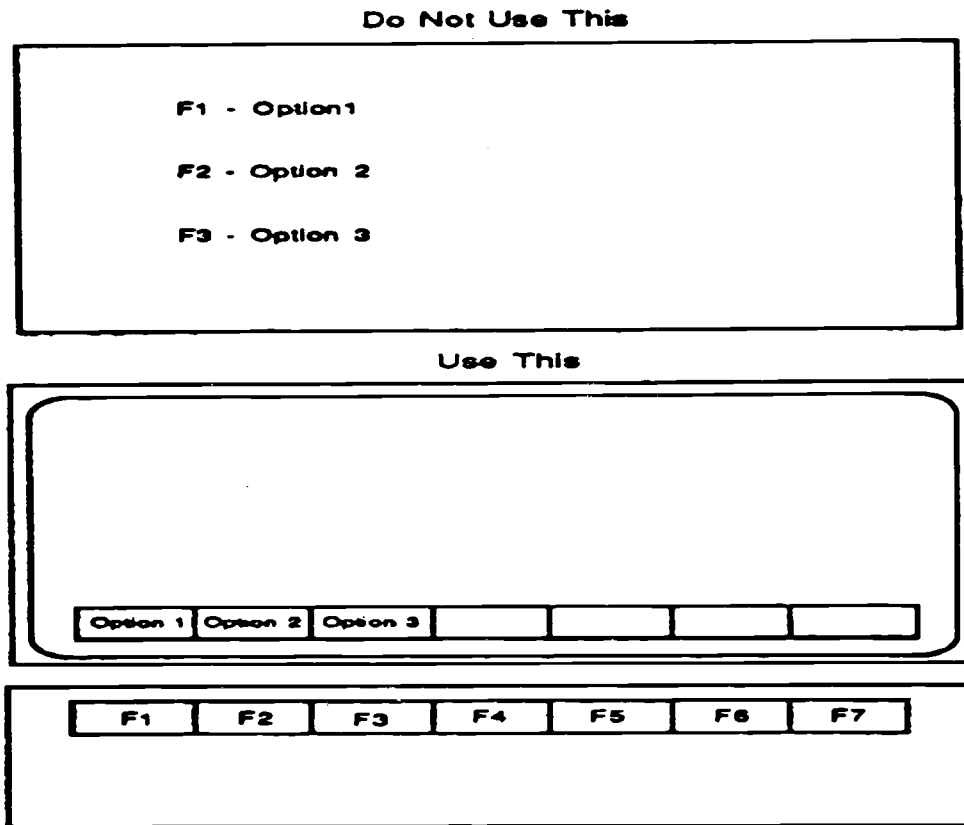


FIGURE 8.9: Recommended Location for Function Key Labels

keyboard, just below the corresponding labels.

8.4.6 SINGLE KEYING

8.4.6.1 Single Activation of Function Keys

Ensure any key will perform its labeled function with a single activation and will not change function with repeated activation without indicating the new function or change in mode.

8.4.6.2 Single Key for Continuous Functions

When a function is continuously available, assign that function to a single key.

8.4.6.3 Single Keying for Frequent Functions

Keys controlling frequently used functions should allow single key action and should not require double (control/shift) keying.

REFERENCE LIST

Paragraph

References

8.0	Smith and Mosier (1986); Helander (1986); DoD (1989b)
8.1	DoD (1992a)
8.2.1	Kearsley (1988) p. 9
8.2.4.4	Horton (1990) p. 264
8.2.4.7	Brown (1988) p. 167, para 9.27
8.2.4.8	Hurd (1983) Cited in Horton 1990, p. 264
8.2.5	Walker (1987) Cited in Horton 1990
8.2.5.2	Relles and Price (1981) Cited in Horton 1990
8.2.5.3	Relles and Price (1981) Cited in Horton 1990
8.2.5.4	Relles and Price (1981) Cited in Horton 1990
8.2.5.5	Relles and Price (1981) Cited in Horton 1990
8.2.5.6	Relles and Price (1981) Cited in Horton 1990
8.2.6	OSF (1990) p. 8-2, para 8.1.1
8.2.6.1	OSF (1990) p. 8-2, para 8.1.1
8.2.6.6	OSF (1990) p. 8-3, para 8.1.3
8.2.7	Kearsley (1988) p. 79
8.2.7.1	Kearsley (1988) p. 79
8.2.7.2	Kearsley (1988) p. 79
8.2.7.3	Kearsley (1988) p. 79
8.2.9.1	DoD (1991) p. 7-3, para 7.3
8.2.9.2	Otte (1982) p. 273, para 3.7
8.2.9.3	Smith and Mosier (1986) p. 298, para 4.0.5
8.2.9.5	Smith and Mosier (1986) p. 302, para 4.0.17
8.2.9.6	Kearsley (1988) p. 68
8.2.9.6g	Fenchel (1981) Cited in Horton (1990)
8.2.10.1	Kearsley (1988) p. 76
8.2.10.2	Unix System Laboratories (1991) p. 4-27, para 3
8.2.12.1	Kearsley (1988) p. 25
8.2.13.1	Unix System Laboratories (1991) p. 4-27, para 3
8.2.13.7	Nickerson (1986); Smith and Mosier (1986) p. 302, para 4.0.17
8.2.13.8	Dumas (1988) Cited in Horton (1990) p. 264; Rockley (1987) Cited in Horton (1990) p. 264
8.2.14	Kearsley (1988) p. 19
8.2.7.2a	DoD (1991) p. 6-2, para 6.1
8.2.7.2e	Thomas and Carroll (1981)
8.3.1.1	Smith and Mosier (1986) para 3.0-9

REFERENCE LIST (Cont.)

<u>Paragraph</u>	<u>References</u>
8.3.1.2	Lickteig (1989) p.9
8.3.1.3	Williams et al. (1987a) Appendix A p. A-2; Smith and Mosier (1986) para 8.3.7.2-2
8.3.1.4	Smith and Mosier (1986) para 3.0-12
8.3.1.5	Chao (1987) p. 360 and 361
8.3.1.6	Smith and Mosier (1986) para 3.0-17
8.3.1.7	Smith and Mosier (1986) para 3.0-7
8.3.1.8	Lickteig (1989) p. 35
8.3.1.9	Lickteig (1989) p. 34; Williams et al. (1987a) Appendix A p. A-3
8.3.1.10	Smith and Mosier (1986) para 3.0-8
8.3.1.11	Hamel and Clark (1986) p. 29; Smith and Mosier (1986) para 3.0-3
8.3.1.12	Williams et al. (1987b) Appendix A p.A-3
8.3.1.13a	Smith and Mosier (1986) para 3.0-6, 19
8.3.1.13b	Smith and Mosier (1986) para 3.0-16
8.3.1.13c	Smith and Mosier (1986) para 3.0-11
8.3.1.14a	Smith and Mosier (1986) para 3.0-5
8.3.1.14b	Smith and Mosier (1986) para 3.0-2
8.3.1.14c	Smith and Mosier (1986) para 3.0-19
8.3.1.14d	Smith and Mosier (1986) para 3.0-19; Salvendy (1987)
8.3.1.15a	Williams et al. (1987b) Appendix A p. A-3; Hamel and Clark (1986) p. 30; DoD (1989b)
8.3.1.15b	Smith and Mosier (1986) para 3.0
8.3.1.15c	Smith and Mosier (1986) para 3.0-15; Baeker (1980); Hamel and Clark (1986) p. 30; Mallery (1985) p. 26; McCann (1983) p. 4; Slominski and Young (1988) p. 5
8.3.1.15d	Smith (1986) para 1.0-3
8.3.1.16a	DoD (1989a) para 5.15.4.1.1.1; Smith and Mosier (1986) para 3.0-20
8.3.1.16b	DoD (1989a) para 5.15.4.1.1.2
8.3.1.16c	Smith and Mosier (1986) para 3.0-21, DoD (1989a) para 5.15.4.1.1.3
8.3.1.17a	Smith and Mosier (1986) para 3.1-2
8.3.1.17b	Smith and Mosier (1986) para 3.0-18;
8.3.1.18a	Bowser (1991) p. 6; Lewis and Fallesen (1989) p. 94; HFS (1988)

REFERENCE LIST (Cont.)

<u>Paragraph</u>	<u>References</u>
8.3.1.18b	Bowser (1991) p. 6; Lewis and Fallesen (1989) p. 94
8.3.1.18c	Lewis and Fallesen (1989) p. 94
8.3.2.1	Bowser (1991) p. 6; Smith and Mosier (1986) para 3.4-1; Bullinger et al. (1987) pp. 312-3; Hamel and Clark (1986) p. 26-27; Mallery (1985) p. 28, 9 p. 2, 14 p. 6, 3 p. 360
8.3.2.2	Smith and Mosier (1986) para 3.4-2
8.3.2.3	Smith and Mosier (1986) para 3.4-3
8.3.2.4	Smith and Mosier (1986) para 3.4-4
8.3.2.5	Smith and Mosier (1986) para 3.4-7
8.3.2.6	Smith and Mosier (1986) para 3.4-6
8.3.2.7	Smith and Mosier (1986) para 3.4-5
8.3.3.1	Smith and Mosier (1986) para 3.2-12
8.3.3.2	Smith and Mosier (1986) para 3.2-11
8.3.3.3	Smith and Mosier (1986) para 3.2-19
8.3.3.4	Smith and Mosier (1986) para 3.2-8
8.3.3.5	Smith and Mosier (1986) para 3.2-10
8.3.3.6	Smith and Mosier (1986) para 3.2-13
8.3.3.6a	Smith and Mosier (1986) para 3.2-15
8.3.3.6b	Bowser (1991) p. 6; Smith and Mosier (1986) para 3.2-18
8.3.3.6c	Smith and Mosier (1986) para 3.2-14
8.3.3.7a	Smith and Mosier (1986) para 3.2-7
8.3.3.7b	Smith and Mosier (1986) para 3.2-6
8.3.3.8	Smith and Mosier (1986) para 3.2-5
8.3.3.9	Smith and Mosier (1986) para 3.2-2
8.3.4.1	Smith and Mosier (1986) para 3.3-5
8.3.4.2	Smith and Mosier (1986) para 3.3-8
8.3.4.3	Bowser (1991) p. 7; Smith and Mosier (1986) para 3.3-7
8.3.4.4	Baeker (1980); Williams et al. (1987a) Appendix A p. A-2; Smith and Mosier (1986) para 3.3-1
8.3.4.6	Harell (1987) p.5
8.3.4.7	Smith and Mosier (1986) para 3.3-3
8.3.4.8	Smith and Mosier (1986) para 3.3-2
8.3.4.9	Smith and Mosier (1986) para 3.3-4
8.3.4.10	Smith and Mosier (1986) para 3.3-6

REFERENCE LIST (Cont.)

<u>Paragraph</u>	<u>References</u>
8.3.5.1	Smith and Mosier (1986) para 3.5-7 and 3.5-11
8.3.5.2	Smith and Mosier (1986) para 3.5-8
8.3.5.3	Smith and Mosier (1986) para 3.5-4
8.3.5.4	Smith and Mosier (1986) para 3.5-5
8.3.5.5	Smith and Mosier (1986) para 3.5-13
8.3.5.6	Smith and Mosier (1986) para 3.5-6
8.3.5.7	Smith and Mosier (1986) para 3.5-3
8.3.5.8	Smith and Mosier (1986) para 3.5-12
8.3.5.9	Smith and Mosier (1986) para 3.5-9
8.3.5.10	Smith and Mosier (1986) para 3.5-10
8.3.5.11	Smith and Mosier (1986) para 3.5-1
8.3.5.12	Bowser (1991) p. 8
8.3.5.13	Bowser (1991) p. 7
8.3.6.1	Smith and Mosier (1986) para 3.6-5
8.3.6.2	Smith and Mosier (1986) para 3.6-4 and 3.6-3
8.3.6.3	Smith and Mosier (1986) para 3.6-2
8.3.6.4	Smith and Mosier (1986) para 3.6-1
8.4.1.1a	Smith and Mosier (1986) para 3.1.4-2
8.4.1.1b	Smith and Mosier (1986) para 3.1.4-1
8.4.1.1c	Smith and Mosier (1986) para 3.1.4-3
8.4.1.2	Bowser (1991) p. 9; Smith and Mosier (1986) para 3.1.4-10
8.4.1.3	Smith and Mosier (1986) para 3.1.4-12
8.4.1.4	Ziegler and Fähnrich (1988) p. 129
8.4.1.5	Ziegler and Fähnrich (1988) p. 129
8.4.1.6	Ziegler and Fähnrich (1988) p. 129
8.4.1.7	Smith and Mosier (1986) para 3.1.4-11
8.4.1.8	Nielson (1987) p. 248; Bullinger et al. (1987) p. 309; McCann (1983) pp.3-4
8.4.1.9	Smith and Mosier (1986) para 3.1.4-16
8.4.2.1	Smith and Mosier (1986) para 3.1.4-15
8.4.2.2	Smith and Mosier (1986) para 3.1.4-14
8.4.3.1	Smith and Mosier (1986) para 3.1.4-7
8.4.3.2	Smith and Mosier (1986) para 3.1.4-8
8.4.4.1	Smith and Mosier (1986) para 3.1.4-4

REFERENCE LIST (Cont.)

Paragraph

References

8.4.4.2	Smith and Mosier (1986) para 3.2.4-5
8.4.4.3	Sidorsky (1994) p. 1.1-12
8.4.5.1	Smith and Mosier (1986) para 3.1.4-18
8.4.5.2	Smith and Mosier (1986) para 3.1.4-18
8.4.5.3	Smith and Mosier (1986) para 3.1.4-17
8.4.6.1	Smith and Mosier (1986) para 3.1.4-9
8.4.6.2	Smith and Mosier (1986) para 3.1.4-13
8.4.6.3	Smith and Mosier (1986) para 3.1.4

9.0 TEXT

Section 9.1 addresses general topics unique to textual windows (i.e., data entry/update screens) that were not covered in Section 5.0, Windows.

Section 9.2 addresses form filling as an interactive dialog. This approach to data entry requires little or no training and allows a relatively slow system response time. Applications primarily use form filling for completing standard message and data entry forms. As with any aspect of the HCI design, consistency of design is of paramount importance. The guidelines presented deal more with interactive control than with data entry. For more information on data entry, see Smith and Mosier (1986), MIL-STD-1472D (DoD 1989a), or DOD-HDBK-761A (DoD 1989b).

9.1 TEXTUAL WINDOWS

This section addresses general guidelines related to windows that are primarily textual (i.e., data entry/update screens).

9.1.1 DATA FIELD LABELING

In general, the appearance of the data should be pleasing to the eye with the arrangement uncluttered and functionality efficient. The following list of guidelines should help to achieve these objectives:

- a. Displays should not be different from paper forms without justification, and field ordering should be in the logical sequence from the user's point of view.
- b. The layout of data fields should be consistent within an application because one of the overall DoD architecture goals is to have consistency across all DoD applications in the layout of commonly used display (e.g., the "views" presented by applications for querying related databases).
- c. The data field labels should be easily distinguishable from the actual data. This distinction could be accomplished using different fonts for labels and data, or using special characters as separators. For example, each label should be followed by a colon (:) and separated from the actual data by at least two spaces.

- d. Columnar data should be distinctly separated (at least three spaces between columns) with column headings displayed above the data and at least one row separating the column heading and the data.
- e. Labels should be consistent throughout an application or set of applications.
- f. In ordinary use, field labels should be protected and transparent to keyboard control so the cursor skips over them when spacing or tabbing.
- g. When a dimensional unit (e.g., \$) is always associated with a field, it should be displayed as part of the label so entry is not required by the user.

9.1.2 UPDATABLE FIELDS

Guidelines for data field updates follow:

- a. Updatable fields should be distinguished by underscores below the data field. If highlights or colors are also used, they should be the same throughout an application or set of applications.
- b. Cues should distinguish required from optional fields and should be consistent throughout an application or set of applications.
- c. When the length of a field is variable, the user should not have to right or left justify or remove blanks from the entered data.
- d. The user should be able to enter data in familiar units. The application should perform any required conversions (e.g., between geographic, geodetic, and Military Grid Reference System coordinates).
- e. Authorized personnel should be able to selectively inhibit updatable fields in a multi-field display. Such a feature would allow trainees to take on increasing database maintenance responsibilities as they learn. It also supports efficient on-line accomplishments of "mass changes" when batch updates are not available.

9.1.3 TEXT CURSOR

The purpose of the text cursor is to indicate to the user where entered data will be placed. The text cursor can be in any updatable input field. Guidelines for the text cursor follow:

- a. If the user clicks on a non-updatable field or anywhere on the form, the text cursor should not move.
- b. The text cursor should move between and within fields with the mouse or by using the Return/Enter key, Tab key, or the arrow keys.
- c. The cursor should not obscure the character displayed in the position it designates except for password and other non-display fields.
- d. When in insert mode, the text cursor should appear between the characters where the inserted text will be placed.
- e. When in overwrite mode, the text cursor should highlight the character that will be replaced.

9.2 FORM FILLING

9.2.1 GENERAL

9.2.1.1 Usage

- a. Form-filling dialog should be used as an aid for composing complex control entries.
- b. Form-filling dialog should be used as a means of displaying default values for the parameters in complex control entries
- c. Form-filling dialog should be used for tasks where some flexibility in data entry is needed (such as the inclusion of optional as well as required items), where users will have moderate training, and/or where computer response may be slow.

9.2.1.2 Interrupts for Multiple Entries

Where forms have multiple entries, provide the user BACKUP, CANCEL, and RESTART capabilities for editing the form prior to final input into the system. See Figure 9.1.

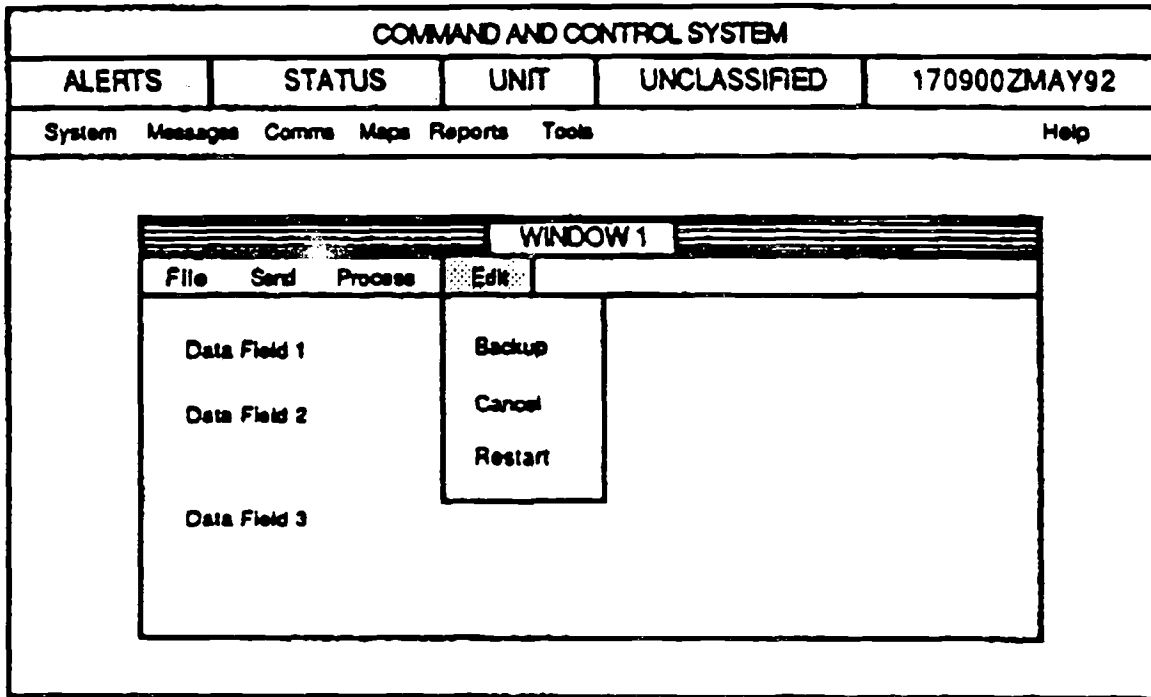


FIGURE 9.1: Example of Interrupt Capability for Multiple Entries

9.2.1.3 Explicit Data Entry

Data entry should be accomplished through an explicit action, such as pressing the ENTER key.

9.2.2 DEFAULTS

9.2.2.1 Automatic Display of Default Data

If default values are used in data entry fields, display them automatically in the appropriate data entry field.

9.2.2.2 Replacement of Default Values

When the user replaces a default value in a data entry field, ensure the default definition is not changed.

9.2.3 CONSISTENCY

9.2.3.1 Consistent Format for Control Forms

Ensure forms for control entry are consistent in format.

9.2.3.2 Format of Form and Hard Copy

When the user is entering data in a computer from a hard copy, the computer form and hard-copy format should be identical, as illustrated in Figure 9.2.

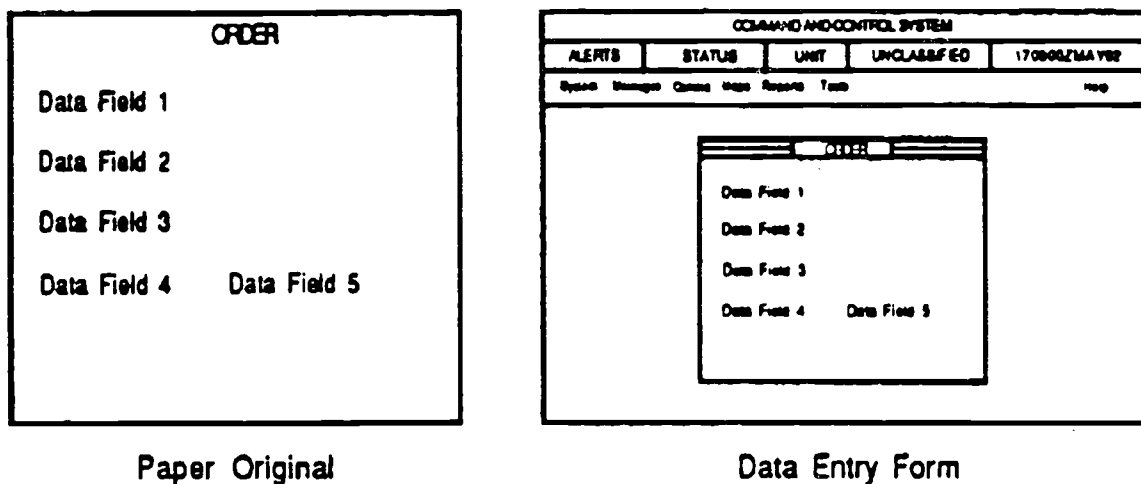


FIGURE 9.2: Example of How a Paper Entry Form and a Computer Data Entry Form Should Be Consistent

9.2.3.3 Entry Dialog Consistency

Dialog strategies for entering words and numbers should be consistent for a given set of logical functions throughout the system.

9.2.3.4 Standard Formats

Data and/or processes that have standard information requirements need to provide the standard format as part of the data screen. Message formats

should include a template for the standard format. Using data entry screens that do not conform to user-accepted format will confuse users.

9.2.4 CURSOR MOVEMENT

9.2.4.1 Cursor Movement Into Non-Data Area

Applications should not allow the user to move the cursor into a non-data entry area during form filling.

9.2.4.2 Convenient Cursor Movement

Ensure the user has a convenient method for cursor control, such as the use of the tab, enter key, or pointing device.

9.2.4.3 Cursor Movement by Explicit Action

When moving from one data entry field to another, the user should be required to take an explicit action, such as hitting the tab control. The software should not automatically advance to the next field.

9.2.4.4 Cursor/Pointing Device Interaction

Pointing device-to-cursor movement ratio should be close to 1:1. If appropriate, the user should be able to select the movement ratio.

9.2.4.5 Initial Cursor Location

When the user first calls up a form, the cursor should be positioned in the first character space of the first data entry field. See Figure 9.3.

9.2.5 DATA FIELD

9.2.5.1 Variable Data Field Format

For data entry fields with variable lengths, the software should automatically justify or truncate the data for the user. No leading characters should be required. See the example in Figure 9.4.

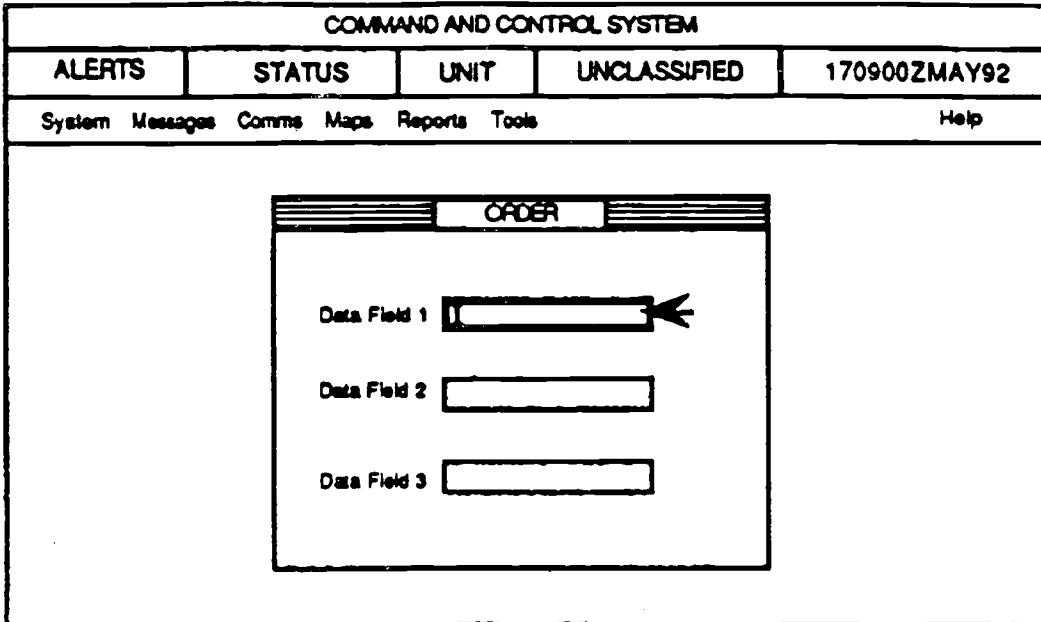


FIGURE 9.3: Cursor Should Appear in the First Character Space of the First Data Entry Field

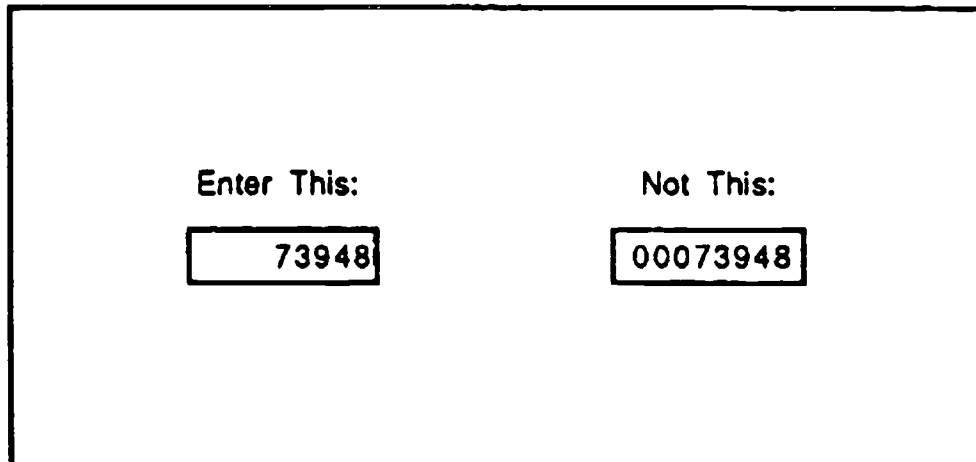


FIGURE 9.4: Data Entry Should Not Require Leading Zeros

9.2.5.2 Consistent Format of Data Fields

The format of data fields used frequently on different forms within and among applications should be consistent from one display to another and should use a format convention consistent with the user's expectations.

9.2.5.3 Computer Identification of Data Omission

When the user omits data, the software should present a message bringing attention to the missing data. The user should either be able to immediately insert the missing data or to identify with a special symbol that the data entry will be deferred.

9.2.5.4 Subgroups Within a Data Field

For data fields longer than 5 to 7 characters, break the field into subgroups of 3 to 4 characters that are separated by a space or delimiter. This should follow a convention consistent with the user's expectations (i.e., names, addresses, some descriptive information should not be subdivided).

9.2.5.5 Data Field Boundaries

Data fields should have distinctly marked boundaries.

9.2.5.6 Data Field Identification

Data entry fields should be clearly identified. Because the interface designs are often complex, users need a positive visual means to identify data entry fields.

9.2.5.7 Field Length

Data entry fields should be of fixed length, with cues given for their length, as illustrated in Figure 9.5.

9.2.5.8 Overwriting

Data entry should not require overwriting of existing or default information. The field should either be empty, or the user should be required to perform an explicit control entry to erase the default data.

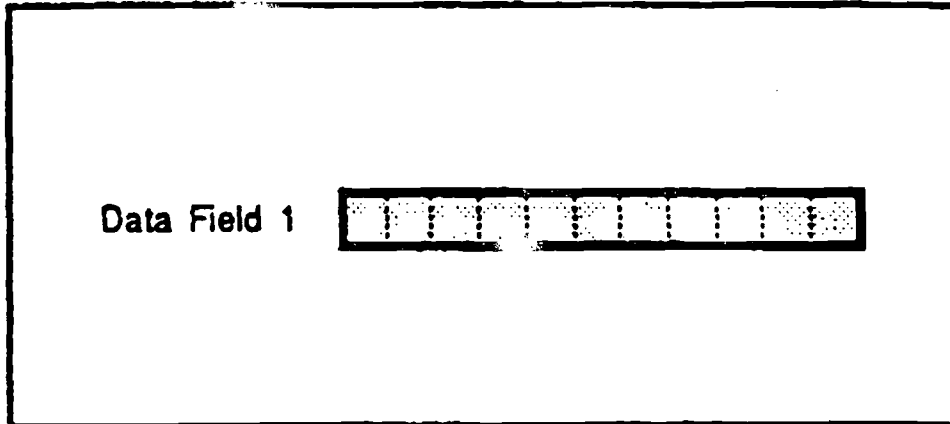


FIGURE 9.5: Visual Cues for Field Length

9.2.5.9 Numeric Data Fields

Numeric data in decimal format should use the decimal as part of the data display. Care should be taken to ensure the field size is adequate for the data range.

9.2.6 ERROR MANAGEMENT

9.2.6.1 Error Correction for Characters and Fields

Ensure the user can easily correct errors on a character-by-character and field-by-field basis.

9.2.6.2 Error Messages

Ensure the software provides understandable error messages to the user when an unacceptable value is entered in a data field.

9.2.7 FORM LAYOUT

9.2.7.1 Multiscreen Form Numbering

If multiscreens are used for a transaction, provide page numbers for each screen also provide a means for rapidly returning to any page, as illustrated in Figure 9.6.

9.2.7.2 Logical Grouping of Data Fields

Related data fields should be grouped together on the same form.

9.2.7.3 Explanatory Messages for Data Fields

Provide explanatory messages for data fields that become visible when the cursor is placed in a field, when a user queries a field by clicking on the title, or by a context-sensitive help system. See the example in Figure 9.7.

9.2.7.4 Distinguishing Data Fields from Other Information

Messages and instructions on a form should be distinguished from data entry fields through consistent location or other means of highlighting. Refer to Figure 9.7.

9.2.7.5 Spacing and Boundaries

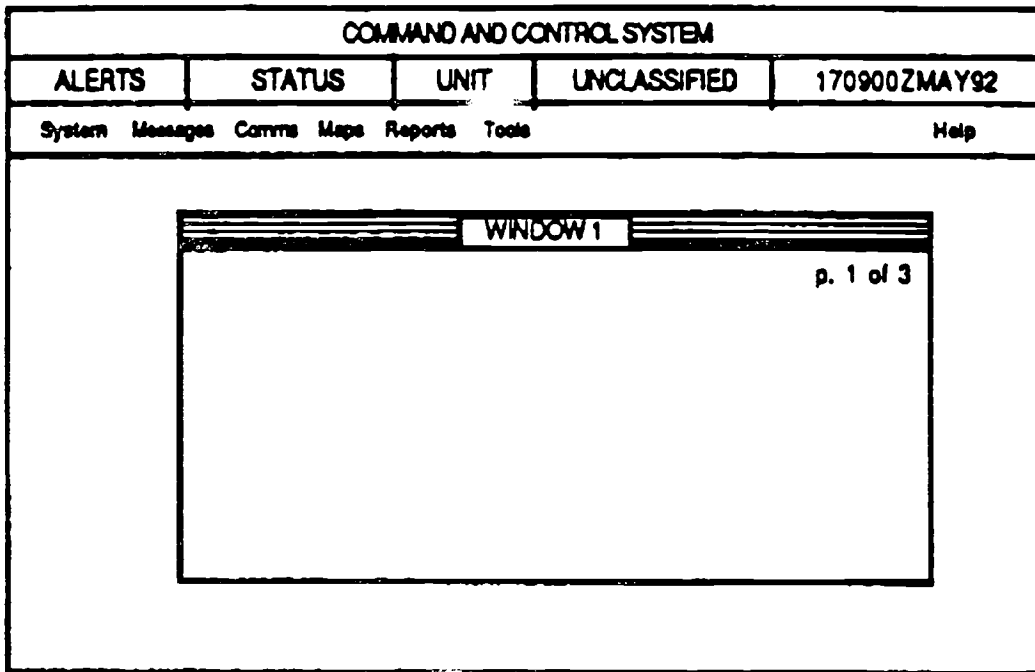
Each data field should have visible space and boundaries between it and other fields.

9.2.7.6 Grouping and Sequencing Fields

Data entry fields should be grouped and ordered on the form in a way that is logical for the task to be performed. This can be by sequence, frequency, or importance.

9.2.7.7 Form Title

Each form-filling dialog display page should have a meaningful title located at the top of the form, as illustrated in Figure 9.8.



OR

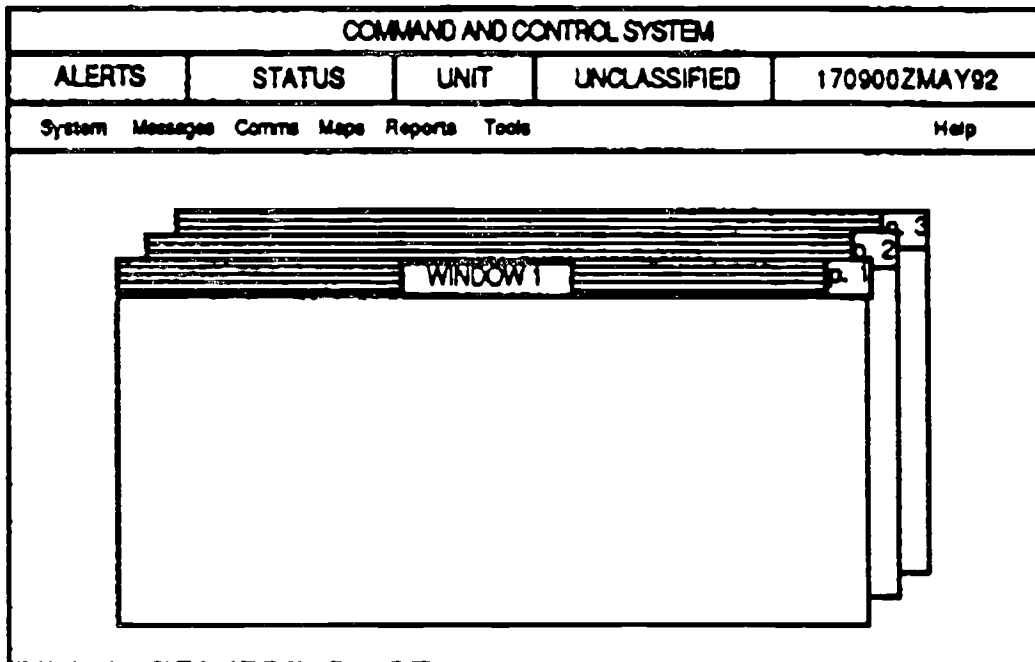
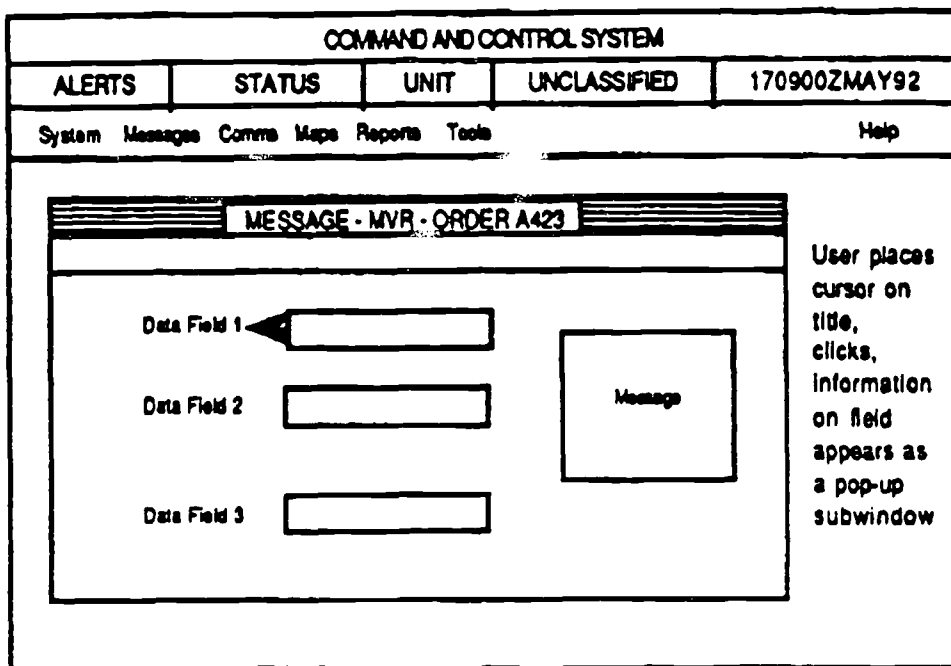


FIGURE 9.6: Example of How Multiple Screens Can Be Numbered



OR

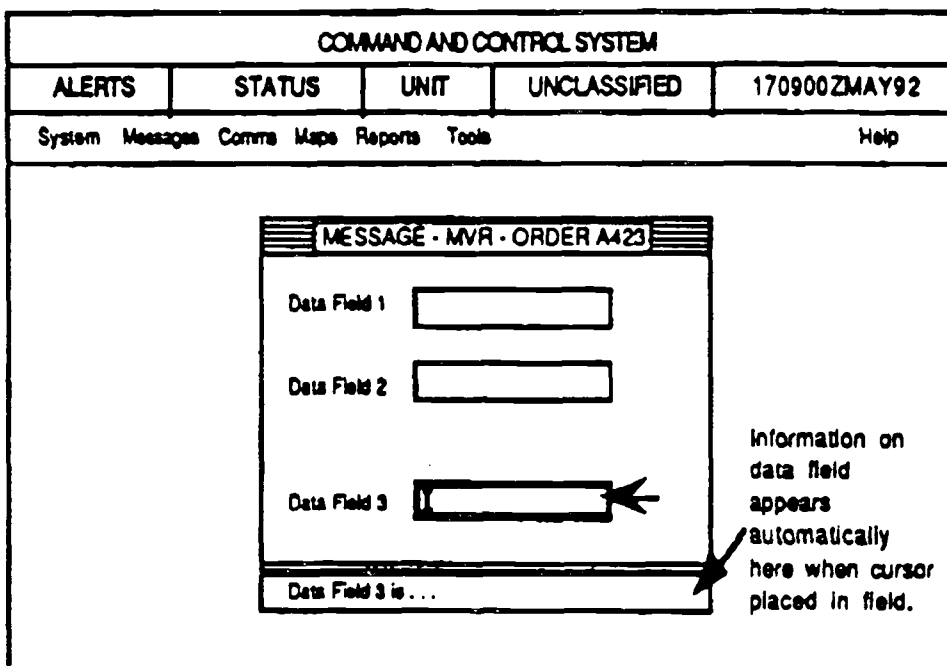


FIGURE 9.7: Example of How Explanatory Messages Can Be Provided

COMMAND AND CONTROL SYSTEM								
ALERTS	STATUS	UNIT	UNCLASSIFIED	170900ZMAY92				
System	Messages	Comms	Maps	Reports Tools				
				Help				
<table border="1"> <thead> <tr> <th colspan="2">MESSAGE - MVR - ORDER A423</th> </tr> </thead> <tbody> <tr> <td colspan="2"></td> </tr> </tbody> </table>					MESSAGE - MVR - ORDER A423			
MESSAGE - MVR - ORDER A423								

FIGURE 9.8: Example of a Form Title

9.2.7.8 Optional Field Labels

Optional fields should be labeled or coded in a readily apparent manner. See Figure 9.9.

9.2.7.9 Optional Field Defaults

When a data entry field in a form is optional, any value displayed in that field should be a default value. See Paragraph 9.2.5.7.

9.2.7.10 Mandatory Fields

The software application should not allow the user to bypass a mandatory field without data entry. See Paragraph 9.2.5.3.

COMMAND AND CONTROL SYSTEM				
ALERTS	STATUS	UNIT	UNCLASSIFIED	170900ZMAY92
System	Messages	Comms	Maps	Reports Tools Help
MESSAGE - MVR - ORDER A423				
Data Field 1	(optional)	<input type="text"/>		
Data Field 2		<input type="text"/>		
Data Field 3		<input type="text"/>		

FIGURE 9.9: Example of an Indication of an Optional Field

9.2.8 LABELING

9.2.8.1 Distinctive Labeling

Data fields, unless similar or identical, should have distinctive, explicitly descriptive labels.

9.2.8.2 Data Field Label Location

Application data entry field labels should be located either directly to the left or above the actual entry field and separated by at least one character.

9.2.8.3 Similar Data Field Labeling

Similar data entry fields should be labeled and located consistently for all forms.

9.2.8.4 Consistent Labels

Labels and instructions should be consistent from one application to another within related applications and to the extent possible across all systems.

9.2.8.5 Field Label Familiarity

Labels for data fields should be composed of terms familiar to the user and the task to be performed.

9.2.8.6 Understandable Labeling

Labeling for data fields and instructions should be easily understood by the typical user.

9.2.8.7 Units of Measure

Units of measure should be part of the data entry field label. If measurement units can change, this portion of the label should change automatically when new units are selected.

9.2.8.8 Blanks Versus Nulls

There should be a visible distinction between blanks and nulls in a data field.

REFERENCE LIST

<u>Paragraph</u>	<u>References</u>
9.1	DoD (1992a)
9.2.1.1a	Smith and Mosier (1986) para 3.1.2-2
9.2.1.1b	Smith and Mosier (1986) para 3.1.2-3
9.2.1.1c	Smith and Mosier (1986) para 3.1.2-1
9.2.1.2	Chao (1986) p. 13
9.2.1.3	Chao (1986) p. 13
9.2.2	Chao (1986) p. 13
9.2.3.1	Smith and Mosier (1986) para 3.1.2-4
9.2.3.2	Sidorsky (1984) p. 1.1-11; Chao (1986) p. 13
9.2.3.3	Lewis and Fallesen (1989) p. 8; Chao (1986) p. 13
9.2.3.4	Bowser (1991) p. 8
9.2.4.1	Chao (1986) p. 13
9.2.4.2	Shneiderman (1988) p. 702
9.2.4.3	Chao (1986) p. 13
9.2.4.4	Chao (1986) p. 13
9.2.5	Chao (1986) p. 13
9.2.5.9	Bowser (1991) p. 8
9.2.6	Shneiderman (1988) p. 702
9.2.7.1	Chao (1986) p. 13
9.2.7.2	Chao (1986) p. 13
9.2.7.3	Chao (1986) p. 703
9.2.7.4	Chao (1986) p. 13
9.2.7.5	Shneiderman (1988) p. 702
9.2.7.6	Shneiderman (1988) p. 702
9.2.7.7	Shneiderman (1988) p. 702
9.2.7.8	Shneiderman (1988) p. 703; Chao (1986) p. 13
9.2.7.9	Chao (1986) p. 13
9.2.7.10	Avery et al. (1990) p. 3-19
9.2.8.1	Chao (1986) p. 13
9.2.8.2	Chao (1986) p. 13
9.2.8.3	Chao (1986) p. 13
9.2.8.4	Shneiderman (1988) p. 702
9.2.8.5	Shneiderman (1988) p. 702
9.2.8.6	Shneiderman (1988) p. 702
9.2.8.7	Chao (1986) p. 13
9.2.8.8	Chao (1986) p. 13

10.0 GRAPHICS

Graphical presentation of data is a critical feature of many emerging DoD applications. This section provides guidelines for presenting data in graphical formats. The applications discussed here include tactical graphics (overlays, symbology, and terrain representation) and pictographic representations (digitized maps, pictures, etc.). Guidelines pertaining to graphical characteristics of the user interface (e.g., screen design, windows, icons, buttons, etc.) are presented in other sections of this document.

Most of the guidelines presented in this section were obtained from Lewis and Fallesen (1989) and Smith and Mosier (1986), who included information gathered from relevant Military Standards and other key documents. Additional guideline materials were obtained through literature reviews.

Section 10.1 focuses on map graphics. The designer of map graphic displays should be aware of the following overarching guidelines that are relevant to electronic map displays:

- The design of maps, including the use of symbology, should be consistent with the user's expectations.
- The level of detail should be consistent with the operational need. Too much or too little detail limits the usefulness of the map.
- Map graphics should have tools built in that allow the user to move easily around the map, to include zooming, panning, insets, registration, and keys for scale.

Section 10.2 focuses on presentation graphics. The goal of presentation graphics is to communicate effectively to the user. The idea, information, or concept communicated should be clear and unambiguous when presented in a visual form, otherwise alternate communication modes should be used. Some emerging technological capabilities allow the direct manipulation of elements of graphic objects within an application. These capabilities should be included in applications designed to interactively create graphics. The *Style Guide* will address three aspects of presentation graphics: graphs, pictures, and diagrams.

10.1 MAPS AND SITUATION DISPLAYS

10.1.1 GENERAL

Maps refer to projected representations of geographic data, usually on flat surface displays. Maps include both natural and man-made features and text and/or graphics and colors used to describe or code those features. Situation displays provide a means of relating changing conditions or events to geographic features represented on maps. Figure 10.1 illustrates a typical map graphic display.

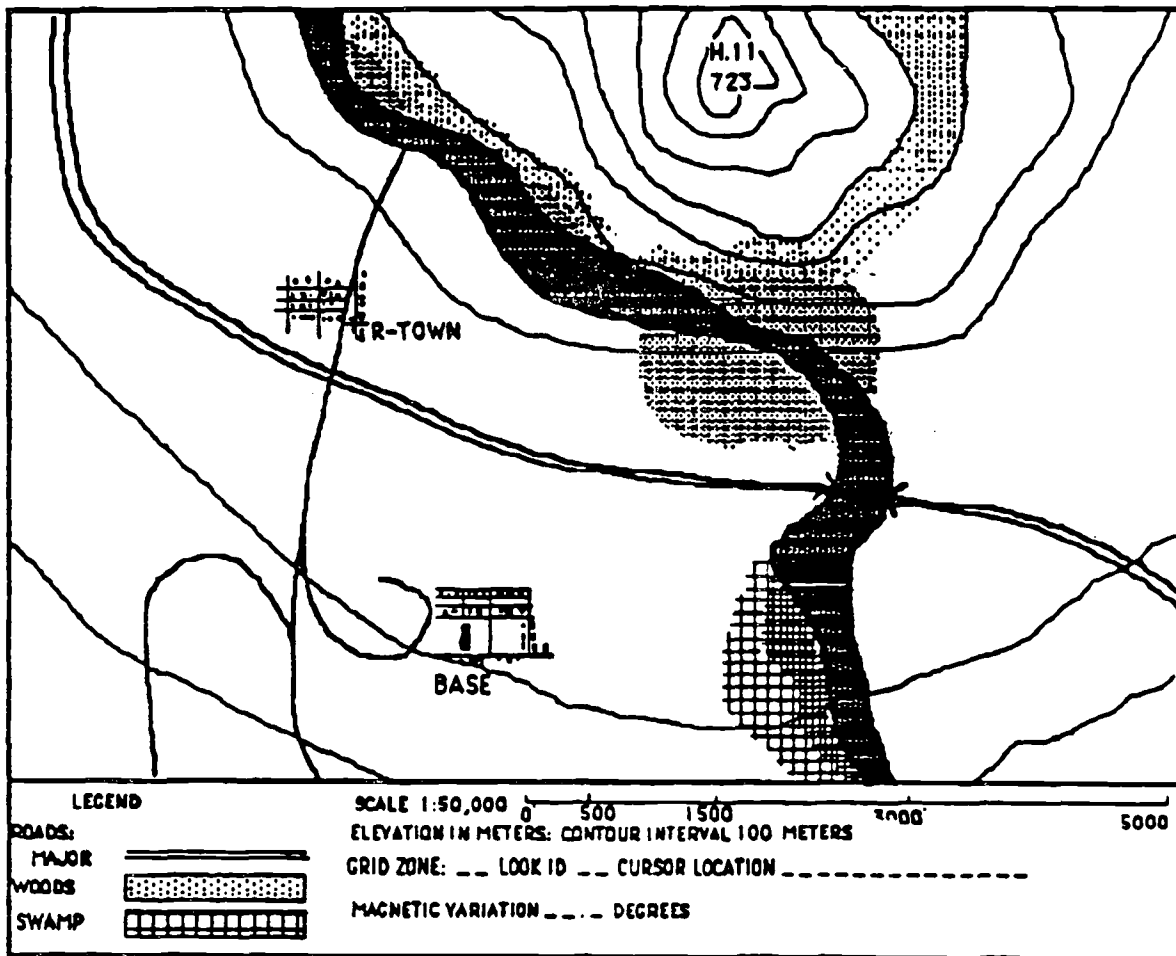


FIGURE 10.1: Typical Electronic Map in Black and White

10.1.1.1 Curvature

Be consistent in projecting the earth's curvature on flat surface maps when displaying large geographic areas.

10.1.1.2 Situation Display Presentation

Provide a means of presenting situation displays as overlays on related map backgrounds.

10.1.1.3 Map Label Position

Position map labels consistently (e.g., beneath or within the feature). Label all significant features without cluttering the display where possible.

10.1.1.4 Map Orientation

Use a consistent map orientation when more than one map will be displayed (e.g., north consistent for all maps). It is recommended that all maps should be north-oriented, and the north direction should be annotated (Figure 10.2).

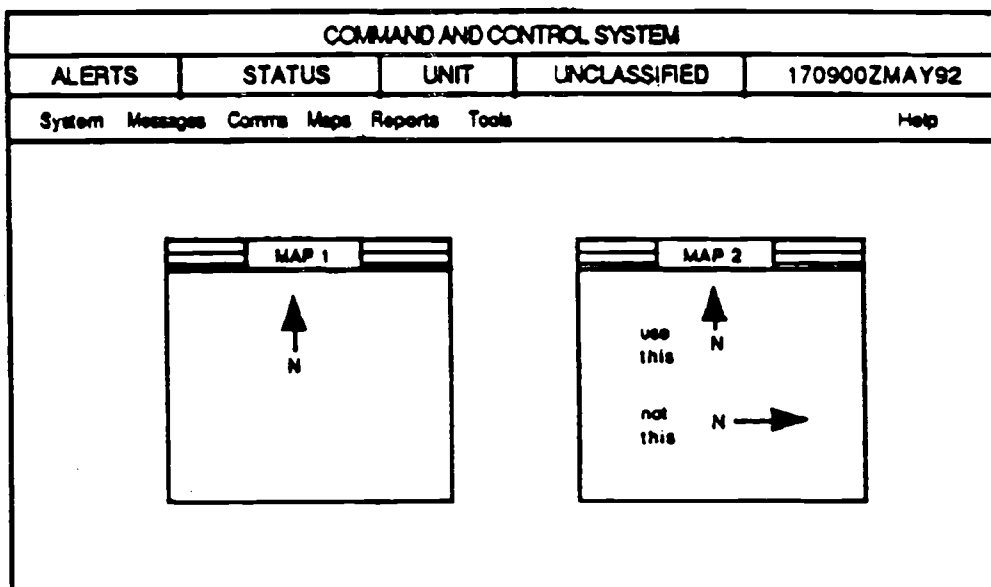


FIGURE 10.2: Example of Consistent Map Orientation

10.1.1.5 Designating Map Areas

Consider using color, shading, texture patterns, or highlighting to define map areas of special interest. Shades (tones) of a single color are preferable to multiple colors when observers must make relative comparisons between or among areas. When using shades of color or texture patterns, the gradation of shades from dark to light should correspond to variation in the variable that is represented (see Paragraph 4.3.3).

10.1.1.6 Automated Tools

Provide automated tools for complex map analyses. The specific tools should be based on the user's needs. For example, avenue of approach, line-of-sight, and trafficability are needed by some but not all users. The user requirements should be determined and appropriate tools provided.

10.1.1.7 Selectability

The user should be able to select a single item within a densely packed group. When a graphics item is selected, it should be highlighted.

10.1.2 STATIC DISPLAY ATTRIBUTES

10.1.2.1 Coverage Area and Resolution

As a minimum, maps must cover the areas of responsibility of the user at each organizational level and provide all essential details required to conduct operations. Map displays should be large enough to permit the simultaneous presentation and visual integration of information required by the user. Small electronic displays may be panned and zoomed (see Section 10.1.3) to increase map coverage. However, at present, such displays have significant visual limitations when compared to traditional, large-format, paper maps.

- a. All critical map features must be represented.
- b. Labels must remain legible at all display resolutions.
- c. Provide a means for reducing clutter while preserving essential information.
- d. Given a command and control application, maneuver commanders at each echelon should be able to view their own areas of operation, activities one echelon above and two echelons below, and activities of friendly adjacent

(flanking) units. The activities of adjacent and deep enemy units that oppose displayed friendly forces should also be displayed. See Figure 10.3.

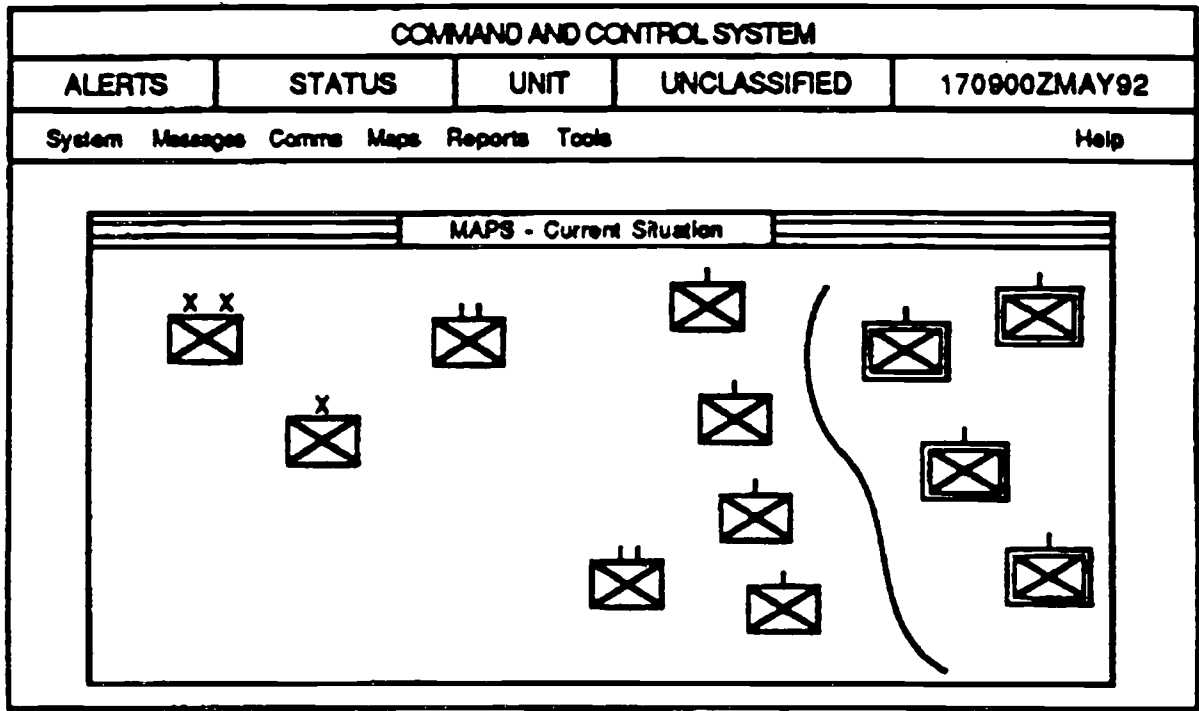


FIGURE 10.3: Brigade's Map Display Showing One Echelon Higher and Two Lower

10.1.2.2 Accuracy of Location

- a. Place symbols accurately on the map or connect to the desired location using arrows, lines, or other pointing devices.
- b. Provide an automated means of registering graphic data with background map information at all display scales.

10.1.2.3 Symbology

Colors, symbols, line size/quality, and fonts should be consistent throughout a given system. Whenever possible, display symbology should conform with published standards (e.g., Army Field Manual 101-5-1 (1985b), North Atlantic Treaty Organization Standardization Agreement 2019 (1990), or the DIA Standard Military Graphics Symbols manual, 1990 draft), but each system should also be able to use a commercial graphics editor to accommodate the

creation and display of system-unique features and symbols. The following guidelines are recommended:

- a. Use standard military symbols in accordance with doctrine when preparing maps and overlays. For example, use the current edition of FM 101-5-1, Operational Terms and Symbols.
- b. Provide a means by which the user may obtain help in identifying unknown symbols or other map information. For example, the user could highlight a symbol and query its meaning through a context-sensitive help feature.
- c. Use standard military map color codes, and provide a user-prompted key defining the color codes that are used. See Section 4.3.
- d. Do not allow map symbols to overlap, particularly if this would obscure their identity. Provide a means for moving background symbols to the foreground or otherwise revealing masked symbols where overlap is unavoidable.
- e. Display essential labels (for example, unit identification) with the symbol; otherwise, provide a means by which the user can display information related to selected symbols. Figure 10.4 illustrates how a user could query a symbol for more detail.
- f. Consider the auxiliary use of alphanumeric coding where graphic data are not already so labeled.
- g. Position symbol labels consistently in accordance with doctrinal guidance.
- h. Digital terrain data available for some versions of electronic map (e-map) allow alternative methods of portraying terrain features. In addition to traditional topographic contour intervals, digital terrain data can present map backgrounds depicting road networks, drainage, vegetation, and soil type. Use shading, coloring, or other visual cues to accentuate terrain features.

10.1.2.4 Location of Displayed Section

A constantly visible display of coordinates associated with the cursor should be displayed in user-selectable coordinate units that can also be conveniently changed where location information is frequently used. The continuous display of location should be augmented with the capability to fix (point on the map) a location to facilitate moving overlay displays.

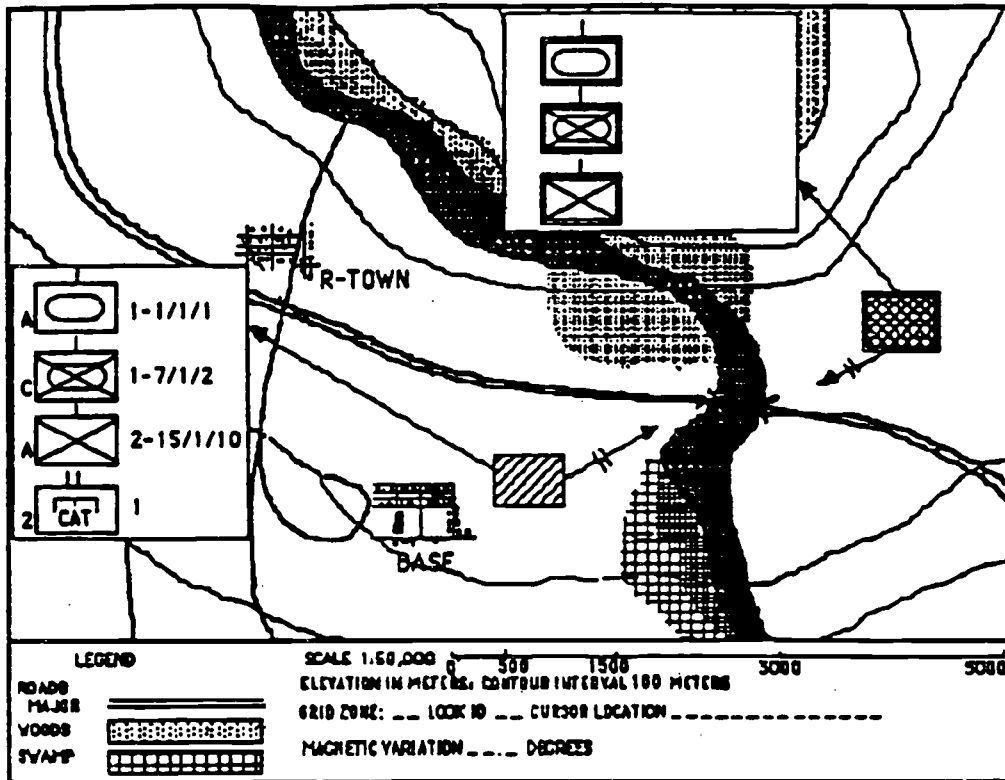


FIGURE 10.4: Querying a Summary Symbol for Detailed Information

- Provide to the user a means of obtaining the exact map coordinates for a selected symbol or map feature by means of querying the symbol or feature. The recommended method of querying an item is to use a pointing device, such as a mouse or trackball cursor.
- When the entire map is not displayed, provide an inset that shows where the displayed portion is located within the larger map (Figure 10.5).
- Provide an automated means for readily determining the distance between points.
- Provide a means for readily determining the bearing between points.

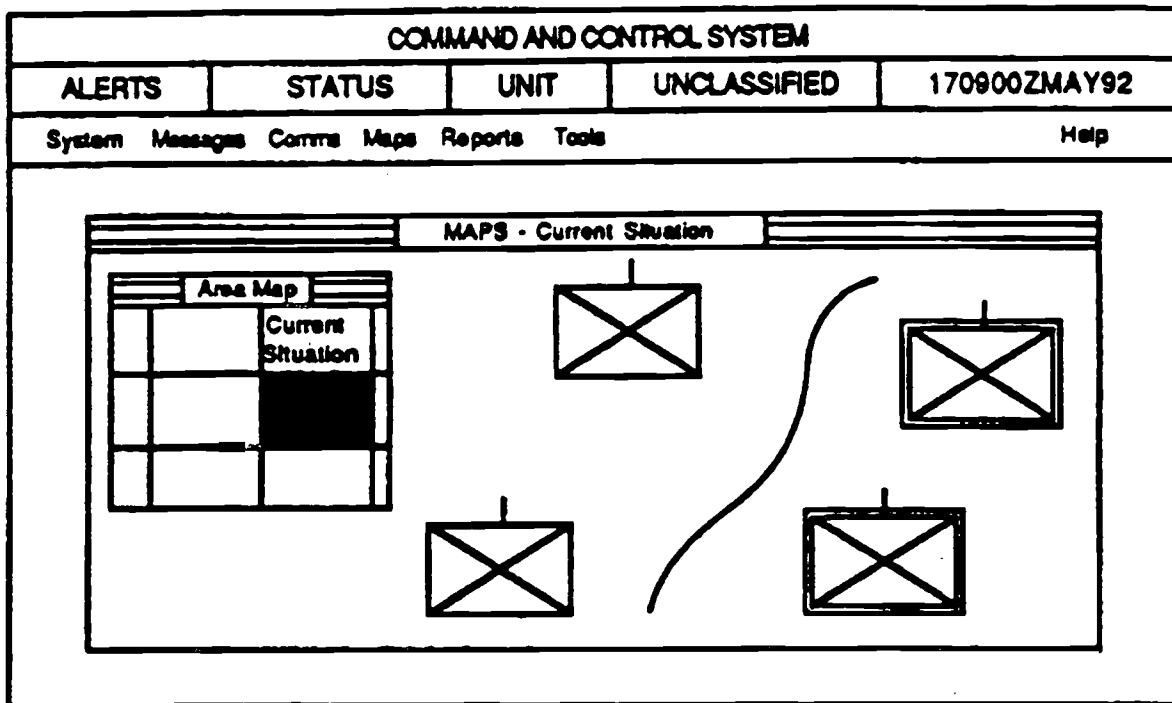


FIGURE 10.5: Example of a Map Inset

10.1.2.5 Area Bounding Boxes

Bounding boxes should be used when displaying maps in the main graphics drawing area. Area bounding boxes are pairs of coordinates defining a rectangular area in terms of latitude and longitude. The bounding coordinates should be displayed for the geographic area being shown.

10.1.3 DYNAMIC CHARACTERISTICS

In a map graphics application, functions should be made available through menus to permit the user to make measurements, perform analysis, and control the appearance of the display. A method is needed to scan and change the scales of the maps because of the limited screen size of many displays. In addition, changes in the tactical situation require updates to various map overlays. The following guidelines should be considered when implementing dynamically changing maps:

10.1.3.1 Panning

- a. Permit the user to change the displayed area by moving a window over the map in any direction. Panning operations may be continuous (preferable) or discrete but should meet the user's requirements.
- b. During panning operations, provide an indicator of position in the overall display.
- c. During panning operations, provide a means for rapidly returning to the starting point.

10.1.3.2 Zooming

- a. Provide a means for moving away from or toward the displayed area (zooming) to obtain a larger view or greater detail.
- b. Zooming must not cause problems in reading symbols, labels, or other map features.
- c. Consider modifying the level of detail (number of symbols and features depicted) to match the degree of zooming used (i.e., more detail for close-up views and less for large-area perspectives).
- d. Of the two methods of zooming (i.e., continuous and discrete), continuous is preferable. Whichever method is used must be satisfactory to the user.
- e. When zooming, symbols should be collapsed into fewer summary symbols to declutter.
- f. Provide a means for quickly returning to the normal display size when zooming.
- g. When changing scales through zooming, provide an indicator that continually shows the appropriate scale.
- h. It is recommended that an inset or window be provided that shows the maximum available map coverage. An example of map coverage (Figure 10.5) would be a graphic square on the inset map that indicates the position of the map currently displayed. In the most useful form, this inset would be interactive and used to set parameters for calling up a screen map display.

10.1.3.3 Automatic Updating

Automatic updating, editing, and distributing map data are among the primary advantages offered by electronic displays. The following guidelines address considerations in implementing these capabilities:

- a. As appropriate, allow the user to select categories of information that will be automatically updated.
- b. Provide stable reference elements (e.g., terrain features, boundaries, etc.) when displays are automatically updated.
- c. Provide a means for readily identifying updates or changes. Critical changes must be easily recognized and distinguishable from other changes to the display. For example, highlight the update until the user acknowledges it.
- d. The user should be permitted to control how often the display is updated and should be able to freeze the display to prevent further updates.
- e. The rate of display update should match the perceptual abilities of the observer to permit successful visual integration of the changing patterns.
- f. Permit the user to freeze the display to prevent further updates. Provide a warning while the automatic display updating is suspended and when resuming automatic updating. Provide an option to either resume at the current time or at the time updating was suspended.

10.1.3.4 Sequencing

Display sequencing may be used to reduce clutter (e.g., presenting map overlays in succession), to reproduce temporal changes in the display database (e.g., changes in the tactical situation), and to aid in visualizing simulated changes in the battlefield situation.

- a. Allow the user to control the rate of sequencing where possible.
- b. Provide a capability to pause or suspend sequencing operations and provide an indicator of the status of sequencing operations.
- c. Allow the user to present sequenced displays in forward or reverse order as appropriate.

- d. Provide a means for the user to return quickly to a selected display within a sequence of displays.
- e. Consider using animation as an aid to the pictorial display for complex objects.

10.1.3.5 Grid Overlay

Provide a user-selectable grid overlay that is keyed to the coordinate system of the map. It should be easy for the user to turn the grid on and off. Coordinate keying of the overlays must be clearly specified and easily operated by the user.

10.1.3.6 Dynamic Map Legend

The map display should have an associated window giving relevant information in a continuous display. The information should include map scale, cursor location, graphic of map coverage, and status (i.e., "working," "computing," "available," etc.).

10.1.3.7 Cursor Design

The cursor should include a point designation feature (e.g., cross hairs or a v-shaped symbol) because fine accuracy is often required in positioning the cursor.

10.1.3.8 Distance/Azimuth

Provide a distance/azimuth function that calculates the distance (range) and azimuth (bearing) between any two selectable points or symbols. Distance should be presented in selectable units (feet, meters, miles, or kilometers). Azimuth should be displayed in degrees from true north.

10.1.3.9 Position Determination

The "determine position" function calculates the position of the point that is identified, and the answer is provided textually by latitude and longitude, distance (in nautical miles), and an azimuth. Coordinates should be presented in a selectable coordinate system (e.g., Universal Transverse Mercator, latitude/longitude, or Military Grid Reference System).

10.1.4 CREATING AND EDITING MAP GRAPHICS

10.1.4.1 Standard Symbol Library

Provide a library of standard symbols and a means of transferring and manipulating symbols.

10.1.4.2 Labeling Symbols

Provide an easy means of labeling symbols. Consider automated means of aiding the user in labeling and enforcing labeling conventions.

10.1.4.3 Building Symbols and Overlays

Provide automated tools to assist the user in constructing new symbols and graphics overlays.

10.1.4.4 Printing Preview

When preparing graphics displays for printing, allow users to preview displays as they will appear when printed.

10.1.4.5 Display Editing

- a. Allow the user to add or delete symbols, labels, or other features without destroying background information.
- b. Allow the user to expand an area of the display as required for accurate placement of critical data.
- c. Provide a means for designating graphic elements for editing. Highlight selected items to provide a visual cue of forthcoming subsequent actions.
- d. Allow the user to reposition selected elements on the display.
- e. Allow the user to remove and restore selected elements.
- f. Allow the user to select from displays of available options when making changes to display attributes (e.g., color, symbols, line types, textures, etc.). Selection should be made by pointing rather than by naming the options.
- g. Provide an easy means for the user to identify attributes currently selected.

- h. Provide the user an easy means to change the attributes of selected graphic elements.
- i. Provide an easy means for naming, storing, and retrieving graphics displays and elements. Also, provide a means for reviewing and selecting from stored graphics files.

10.1.5 MAP DISPLAY CHARACTERISTICS

10.1.5.1 Map as a Base Screen

When an application is map intensive, it is recommended that the map be used as the background or base screen, which should be the maximum display size possible to promote readability.

10.1.5.2 Map Readability

It is beneficial to ensure the readability of map features, since the map is the focus of the user. When possible, the screen design should avoid displays that cover the map, and windows should not obscure the map.

10.1.5.3 Map Cursors

Map cursors should use a crosshair design that has high contrast with the background. It is recommended that cursor size subtend 20 minutes of visual angle so the average user can easily locate it on the map.

10.1.5.4 Graphic Overlays

An overlay is a layer of information (e.g., grids, boundaries, or control measures) that has been drawn on a graphics canvas. Various overlays should be made available to the user to display (make visible), hide from display (make invisible), or delete. The preselection or filtering of graphic overlays is a recommended feature. The decluttering of graphic displays (especially maps) should be assisted.

- a. Review labels and titles used for filters carefully to ensure items are understandable. The filters should be extended to map features, such as roads, cities, vegetation, topography, and political data. The intensity of the map should be controllable to allow fadeout of the map without losing all the map features.
- b. Graphic overlays may overlap map features but should not obscure text

information. The text should be offset with arrows to preserve map legibility.

- c. Include in the graphics package the capability to display a list of available overlays, distinguishing between visible and invisible overlays.
- d. Other possible overlays include boundary lines, oceans, rivers, grids, land masses, railways, and user-generated overlays (created through the graphics editor).
- e. Provide a map overlay editor function.

10.1.5.5 Color Use with Graphic Overlays

Using color to identify symbols is encouraged, but also use redundant coding that does not use color. This caution is especially true for friend-enemy or danger-safe designations. Dots, dashes, shapes, and video effects are recommended. Be careful to avoid visual color illusions caused by color blending (e.g., adjacent red and blue lines are seen as one purple line).

10.2 PRESENTATION GRAPHICS (GRAPHS, PICTURES, AND DIAGRAMS)

Graphs should be used where necessary to visualize relationships among two or more variables, to facilitate comparing sets of data, to aid the observer in visualizing trends in data, and to aid in extrapolating future values of the underlying data. Graphs are also useful when comparing actual data to predicted values, when comparing actual values to established limits in control processes, for representing rapidly changing data, and for interpolating values between known points. In general, graphs have advantages over tabular data in summarizing complex relationships among variables and facilitate information processing and understanding.

Pictures are becoming an increasingly important form of graphic presentation. The multimedia capabilities of developing computer systems have increased the availability of pictures within computer applications. The most frequent command and control picture is a map. The use of scanned maps has transferred the tactical planning focus to the computer interface. The use of pictures on computer screens must be done with great care to avoid misleading the user.

Use diagrams (schematics) when the user requires information concerning the spatial relationship among objects but does not require the level of detail

required by pictures. Schematic representations can be used as an aid to understanding relationships among components of complex systems and as a means of conveying status information concerning the operation of systems and their components. They also provide a medium through which users may manipulate designs and observe consequent actions on modeled systems.

10.2.1 GENERAL

10.2.1.1 Complex Formats

Avoid complex formats, such as three-dimensional presentations and artistic embellishments (pictures, shading, colors, decorative items), which detract from the intended purpose of the graphic.

10.2.1.2 Clarity Preservation

Design graphics to preserve clarity when the graphics must be reproduced or reduced in size. Window sizing should be controlled so no graphic shows partial lines.

10.2.1.3 Appropriateness of Formats

Provide formats (presentation styles) appropriate for the user's level of training and experience. Graphics should utilize user-expected symbols.

10.2.1.4 Data Specific to Task

Provide only those data the user needs for a specific task.

10.2.1.5 Alternative Style Selection

Allow users a selection of alternative presentation styles.

10.2.1.6 Querying Data Elements

Provide a means by which the user can select data elements on the graph and display the associated values.

10.2.1.7 Graphical Versus Tabular

Consider allowing the user to select between graphical and tabular data formats.

10.2.1.8 Consistency

Be consistent in design, format, labels, etc. for each presentation style.

10.2.1.9 Labeling

Clearly label the displayed graphics.

10.2.2 CREATING AND EDITING

10.2.2.1 Computer-Aided Entry

Provide computer aids for the entry and organization of complex graphic data.

10.2.2.2 Data Entry Validation

Validate data entries. The validation may include comparison to a standard range and/or the use of rules for relationships among variables. The validation process should be part of the application software.

10.2.2.3 Data Entry Aids to Plotting

When plotting formats are known, provide templates or other data entry aids to facilitate the entry of graphic data.

10.2.2.4 Automated Plotting of Stored Data

Automate plotting of stored data, and provide the user with automated editing and construction capabilities.

10.2.2.5 Automated Production of Scales

Automate the production of scales, and/or provide the user with automated aids for scaling graphic data.

10.2.2.6 Lines

- a. Provide automated aids for drawing straight and curvilinear line segments.
- b. Use rubberbanding (i.e., provide a visible line that connects a starting point to the current cursor position), which can be made permanent when selected.
- c. Provide automated assistance in joining and intersecting line segments.
- d. Allow the user to identify and select line segments for moving and editing. Typically, this is done through highlighting and dragging the line.
- e. Provide optional, adjustable, grid references to aid the user in aligning horizontal and vertical lines.

10.2.2.7 Rule Specification by the User

Allow the user to specify rules for attributes, relationships, and design, and have the computer apply those rules automatically during the design process. For example, straighten hand-drawn lines, adjust angles between intersecting lines, and complete details of graphic elements.

10.2.2.8 Computer-Aided Drawing

Provide computer-aided methods for drawing figures and a system of prompts or other means to aid the user in the design process.

10.2.2.9 Automatic Scale Reduction

Allow the user to edit or create drawings using a large scale, which will later automatically reduce to the desired scale.

10.2.2.10 Object Manipulation

Provide a basic set of capabilities to resize, copy, move, and rotate displayed objects.

10.2.2.11 Mirror Imaging

Provide a means of producing mirror images (reflecting) as an aid in producing symmetrical graphic displays.

10.2.2.12 Grouping Elements

Permit the user to select and group graphic elements that will be edited in common.

10.2.2.13 Area Fill Capability

Provide an automatic means of filling enclosed areas with selected attributes (e.g., color or texture).

10.2.2.14 Computer Models for Graphical Display Generation

Provide computer models that can generate graphical displays in response to parameters provided by the user. See Figure 10.6.

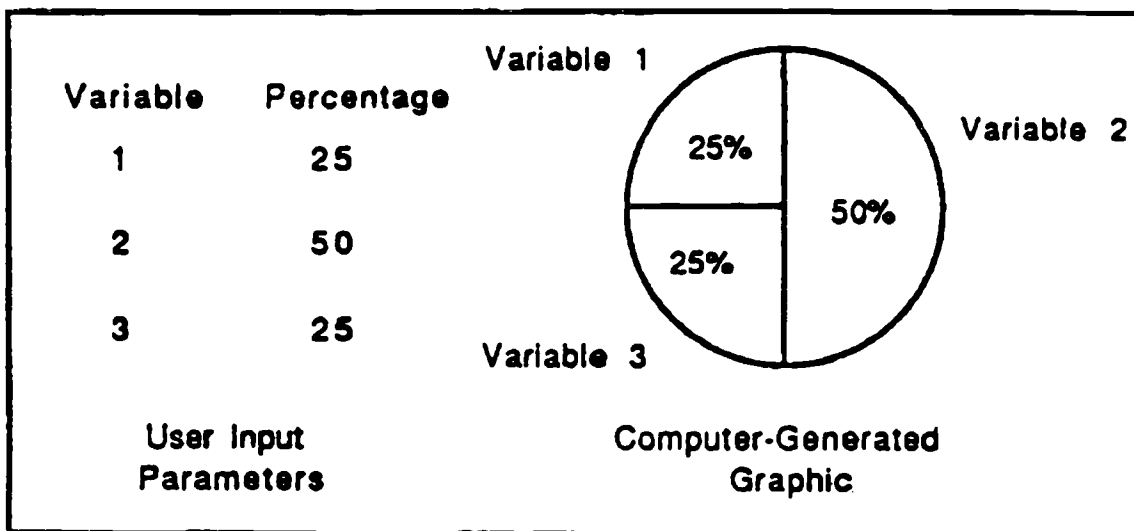


FIGURE 10.6: Example of How a Computer Model Can Generate Graphics from User Input

10.2.3 SCALES, LABELS, AND CODING

10.2.3.1 Standard Scaling Conventions

Use standard scaling conventions: values on the horizontal axis increase to the right of the origin while the values on the vertical axis increase going up from the origin ; independent variables (time or causal events) are plotted against the horizontal axis; and dependent variables (effects) are plotted against the vertical axis.

10.2.3.2 Standard Meanings

Use or assign standard meanings to graphic symbols, and apply them consistently.

10.2.3.3 Color and Pattern Coding

Users prefer colors to patterns for coding lines or filling areas of graphs on visual displays. Good design requires redundant coding be used. See Section 4.3 for color usage guidelines. Printed outputs should use texture for coding, since in most cases color will not be available.

10.2.3.4 Texturing Displays

If texturing must be used, use simple hatching or shading, and avoid patterns

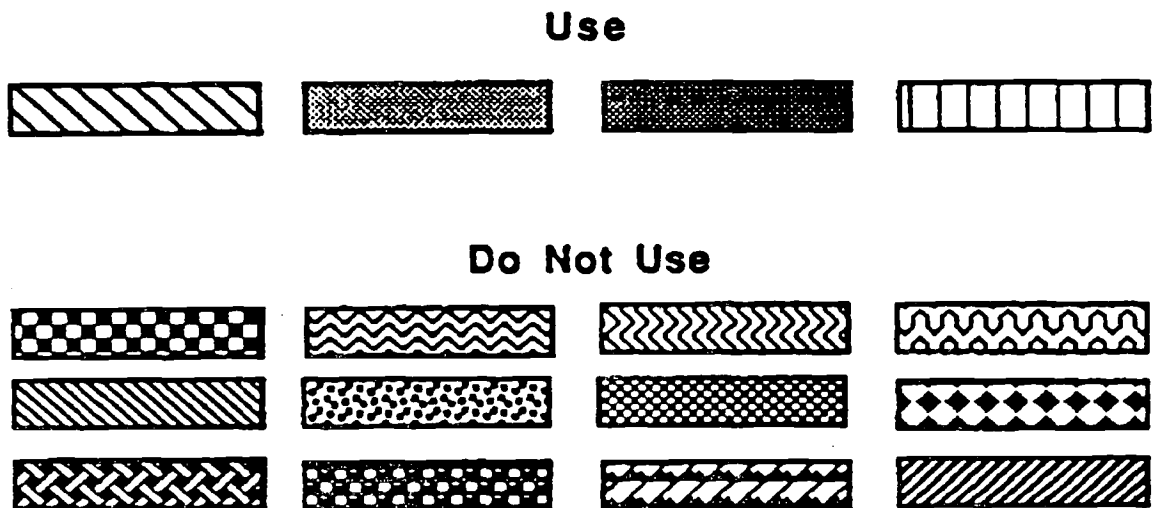


FIGURE 10.7: Texture Patterns

that produce visual illusions of vibration and motion. See Figure 10.7.

10.2.3.5 Axes Breaks in Expanded Scales

When expanding scales to emphasize a limited range of data, provide breaks in

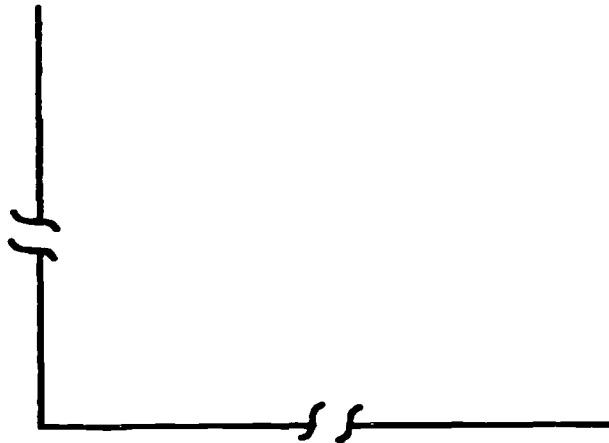


FIGURE 10.8: Example of Breaks in a Graph's Axes When Scales Have Been Expanded

the axes to indicate discontinuities with the origin. See Figure 10.8.

10.2.3.6 Duplicating Axes

When scaled data contain extreme values, it may be difficult for the user to comprehend the scale values in relation to the data. To aid readability, add a copy of the X-axis at the top and a copy of the Y-axis at the right of the graph. Extreme values and data are thus in proximity throughout the graph. In some cases of where numbers of an extremely large or small orders populate the graph, a logarithmic scale may be necessary.

10.2.3.7 Avoid Exaggerated Scales

Avoid the use of exaggerated scales that distort or suppress trends in the data. See Figure 10.9.

10.2.3.8 Formats for Graphic Comparison

Provide identical formats and scales when comparisons are required between

separate graphs, or plot different sets of data on the same graph.

10.2.3.9 Using Linear Scales

Linear scales should be used in preference to non-linear scales wherever

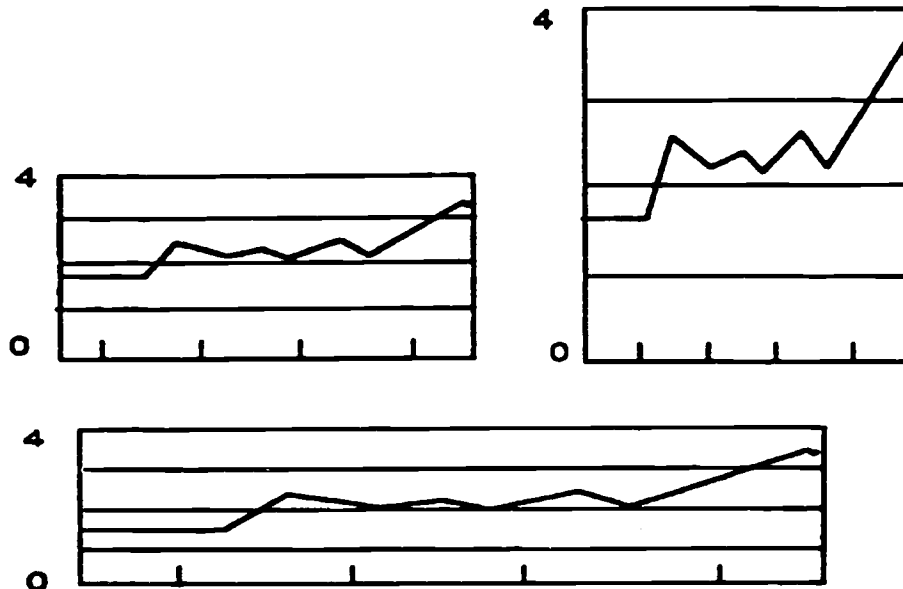


FIGURE 10.9: Example of Distortion Induced in Data Trends by Exaggerated Scales

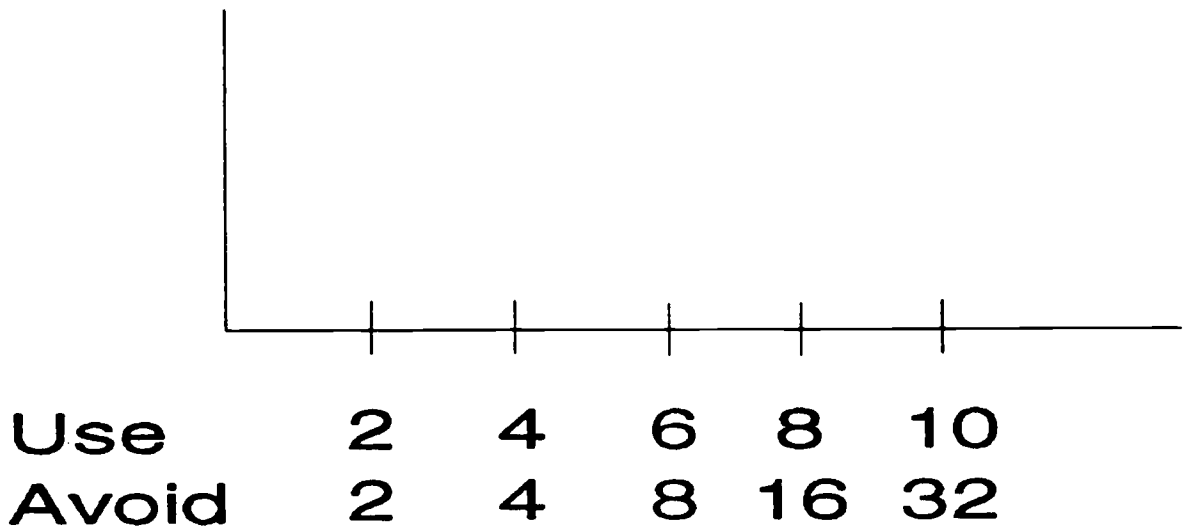


FIGURE 10.10: Example of Linear Versus Non-Linear Scales

practical. For example, see Figure 10.10.

10.2.3.10 Using Logarithmic Scales

Logarithmic scales may be used where comparisons of rates of change and percentages are required or where numbers of both extremely large and extremely small orders of magnitude populate the graph.

10.2.3.11 Multiple Entries

Avoid multiple scales on the axes of a single graph.

10.2.3.12 Labeling Tick Marks

Number or label tick marks corresponding to major scale divisions on the axes, and include a label containing descriptions and units of measurement on each axis.

10.2.3.13 Numbering Scale Divisions

Where practical, use no more than 10-12 major scale divisions separated by up to 9 subdivisions. Whenever the appearance of the display will not be degraded, major scale divisions should be decimal multiples of whole numbers, cover the entire range of the data, and start from zero.

10.2.3.14 Numeric Scale Division

Numeric data scales should begin with zero when users must use displays to compare quantities or different series.

10.2.3.15 Label Format

Labels should use upper and lower case sans serif fonts and be oriented left-to-right for normal reading.

10.2.3.16 Use of Labels

Labels should be used in preference to legends or keys when it is necessary to identify plotted data elements. Labels should be oriented horizontally and located adjacent to the referenced elements. Arrows, lines, or similar pointing conventions may also be used to connect labels to their respective data elements.

10.2.3.17 Location of Legends and Keys

Legends or keys used to identify graphic data elements should be located within the rectangular bounds of the graph, unless such positioning would interfere with interpretation of the displayed data.

10.2.4 IDENTIFYING CRITICAL DATA

10.2.4.1 Displaying Values

Display reference or baseline values when users are required to make comparative evaluations against a fixed standard.

10.2.4.2 Using Supplementary Text

Consider using supplementary text to emphasize features of data requiring user attention. See the example in Figure 10.11.

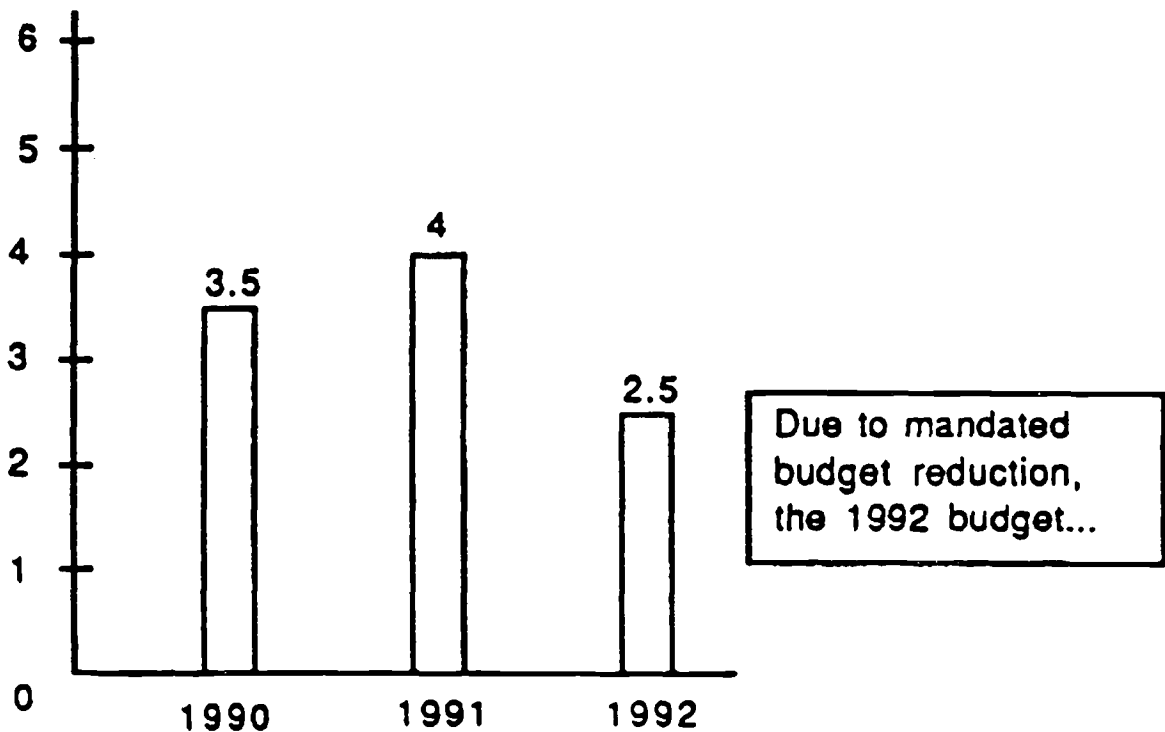


FIGURE 10.11: Example of Supplementary Text and Actual Value Use in a Graph

10.2.4.3 Displaying Data Values with Graphics

Display actual data values in addition to the graphic display where precise readings of values are required, as illustrated in Figure 10.11.

10.2.4.4 Position of Text Used for Labeling

When labeling graphic data, position text consistently with respect to graphic elements.

10.2.5 GRID LINES

A grid is the set of horizontal and vertical lines, including the labeled and scaled axes, which form a rectangular boundary around the graph. Additional horizontal and vertical grid lines corresponding to scale values partition the bounded area of the graph and provide a visual aid in locating and reading points on the graph(s). A grid and grid lines should be used, as appropriate, when presenting data graphically.

10.2.5.1 Grid Line Visibility

Grid lines must be easily distinguishable and should not obscure graphed data.

10.2.5.2 Using Grid Lines

Avoid excessive use of grid lines. Locate grid lines using the guidelines for placement of major scale values. Consider using more grid lines where greater precision is required or where the size of the display will permit their use.

10.2.5.3 User Display of Grid Lines

Where practical, allow the user to determine whether or not grid lines will be displayed.

10.2.6 TYPES OF PRESENTATION GRAPHICS

10.2.6.1 Curve and Line Graphs

- a. Use smoothed curves or straight lines connecting data points (line graphs) when displaying relationships between two continuous variables (e.g., when showing time variation in some quantity).
- b. When a single graph contains multiple curves, designate each curve with an adjacent label. If it is necessary to use a legend, list legend codes in the order in which curves occur in the graphs.
- c. When displaying multiple curves, highlight a curve containing critical data.
- d. Use line coding to distinguish among multiple curves on the same graph, and use coding consistently when the same types of data appear on different displays.
- e. Use a distinct line code (e.g., dashed or dotted lines) when projecting values beyond the actual data set.
- f. For cyclic data, provide at least one full cycle of data.
- g. Consider plotting the difference between two series where comparisons are necessary.

10.2.6.2 Surface Charts

Surface charts provide a means of visualizing the relative contributions of individual elements to the sum of their individual parts, often as a function of time. Figure 10.12 uses a surface chart to illustrate how the total number of items (i.e., the sum of the numbers within each category) varies with time.

- a. Use texture or shading to indicate the area between curves.
- b. Stack the series with the least variable series at the bottom and the most variable at the top.
- c. Place labels within the textured or shaded bands if space is available.

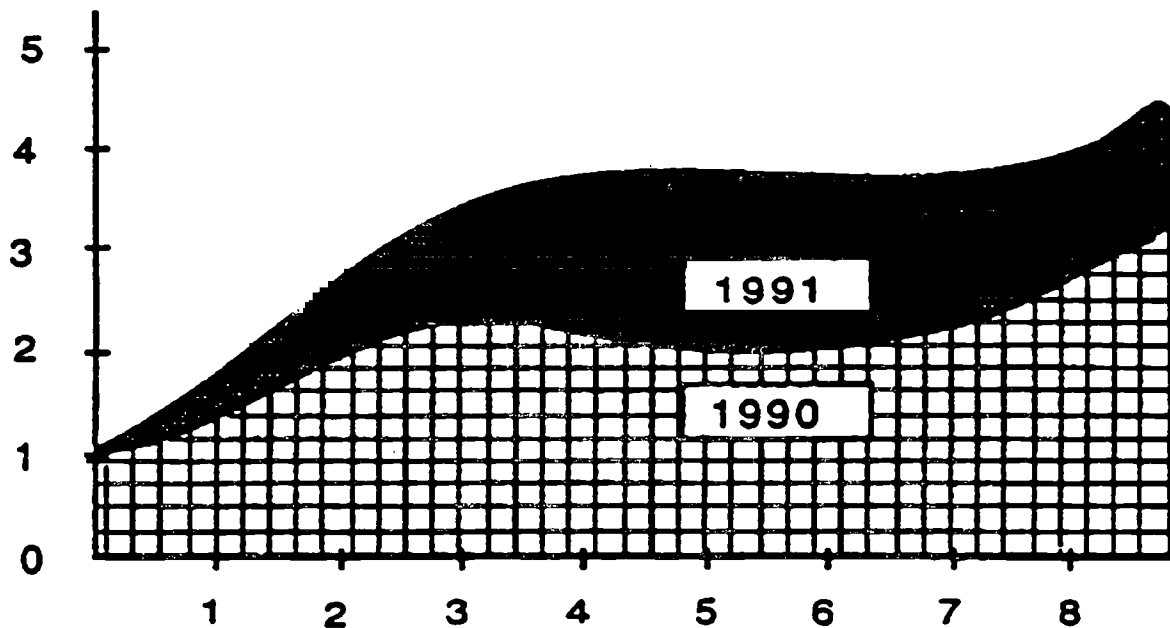


FIGURE 10.12: Example of a Surface Chart

10.2.6.3 Bar Graphs

Bar graphs represent the magnitudes of numeric data by the lengths of parallel bars. Bars may be vertically or horizontally oriented and are usually spaced apart along an axis containing discrete reference points (e.g., months, mid-points of sample intervals, non-numeric categories, etc.). Histograms, or stepcharts, are bar graphs without spacing between bars and are used when a large number of intervals must be plotted. Figure 10.13 illustrates bar graphs. Graphic presentations should be designed to conform to user expectations.

10.2.6.4 Scatterplots

Scatterplots present data as a two-dimensional distribution of points and should be considered when it is necessary to show how variables are related or to represent the spatial distribution of data (for example, impacts on a target). Highlight data points that are particularly significant. Figure 10.14 illustrates a scatterplot.

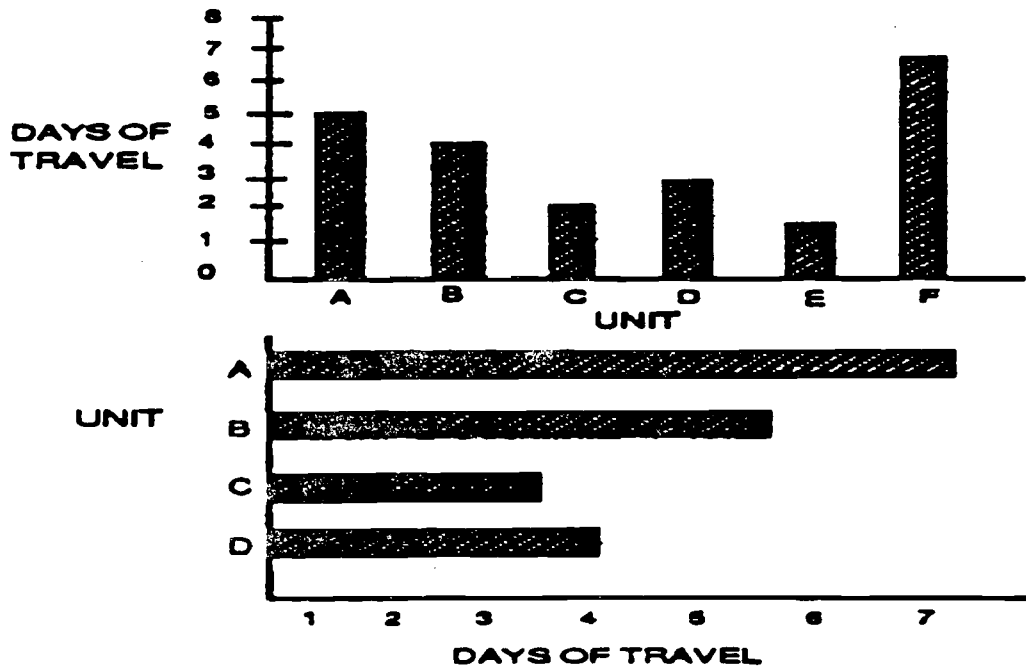


FIGURE 10.13: Examples of Bar Graphs.

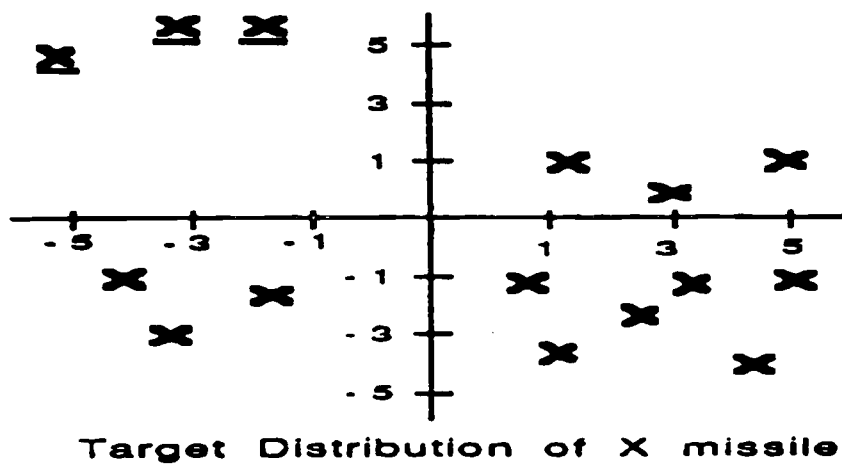


FIGURE 10.14: Example of a Scatterplot

10.2.6.5 Pie Charts

Pie charts, like bar graphs, are used to show the proportional distribution of categories with respect to the sum of the categories. See the example of a pie chart in Figure 10.15.

- a. Place labels in a normal orientation on the segments of pie charts. Segment labels should include numbers that indicate percentages and/or absolute numbers represented by each segment of the display.
- b. Segments requiring emphasis should be highlighted or displaced slightly from the rest of the pie chart, as illustrated in Figure 10.15.

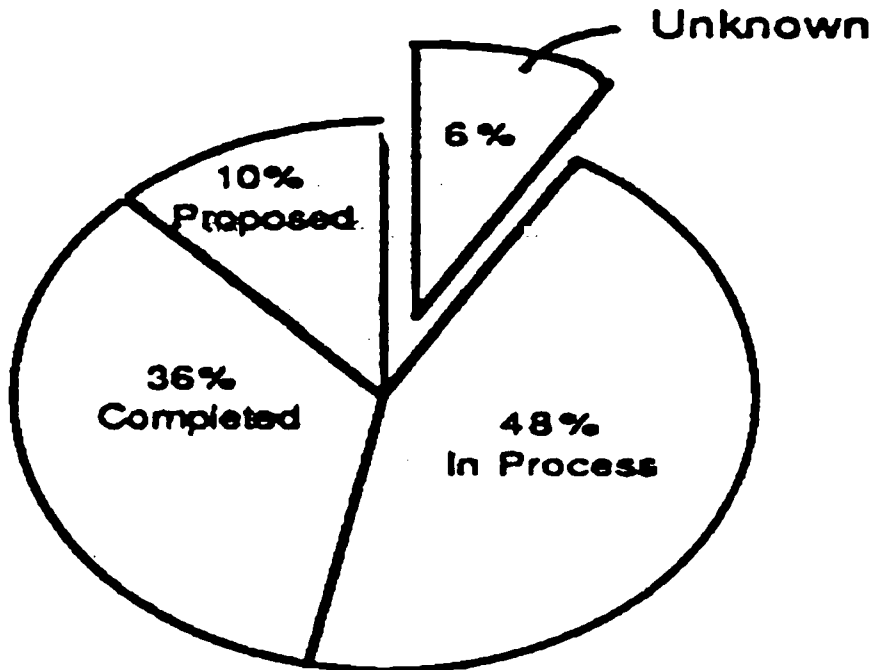


FIGURE 10.15: Example of a Pie Chart

10.2.7 PICTURES

10.2.7.1 Using Pictures

Consider using graphic pictures when a very detailed representation of objects is required. For example, see the scanned map in Figure 10.16.

10.2.7.2 Automated Aids for Pictures

Provide automated aids when users must perform detailed analyses of image data.

10.2.8 DIAGRAMS (SCHEMATICS)

10.2.8.1 Diagrams General

Use diagrams when the user requires information concerning the spatial relationship among objects but does not require the level of detail provided by pictures.

- a. When diagramed data are presented in separate sections, use consistent notations across sections, provide an easy means for users to move among sections, and provide an overview of the entire diagram represented by the individual sections.
- b. Highlight portions of diagrams requiring special user attention.
- c. Provide a capability for the user to rotate displayed diagrams where it is necessary to view the object from different perspectives.

10.2.8.2 Flowcharts

Use flowcharts to provide a schematic representation of sequential processes. Use them also as aids to problem-solving when solutions can be reached by answering a series of questions. Figure 10.17 illustrates a typical flowchart.

- a. As appropriate, sequence flowchart elements in a logical order; otherwise, when designing flowcharts, minimize path lengths to reduce size.

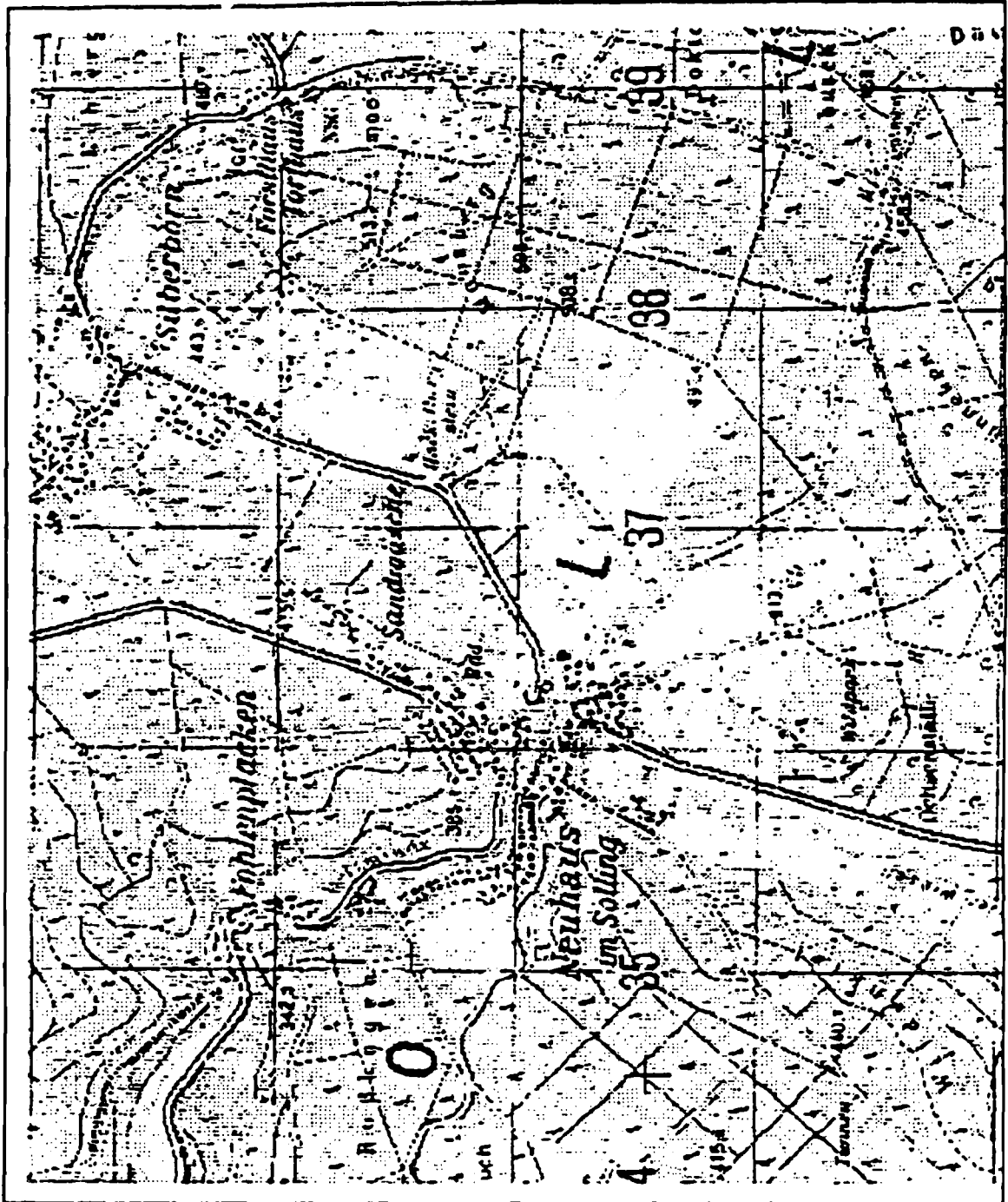


FIGURE 10.16: Example of a Graphic Picture

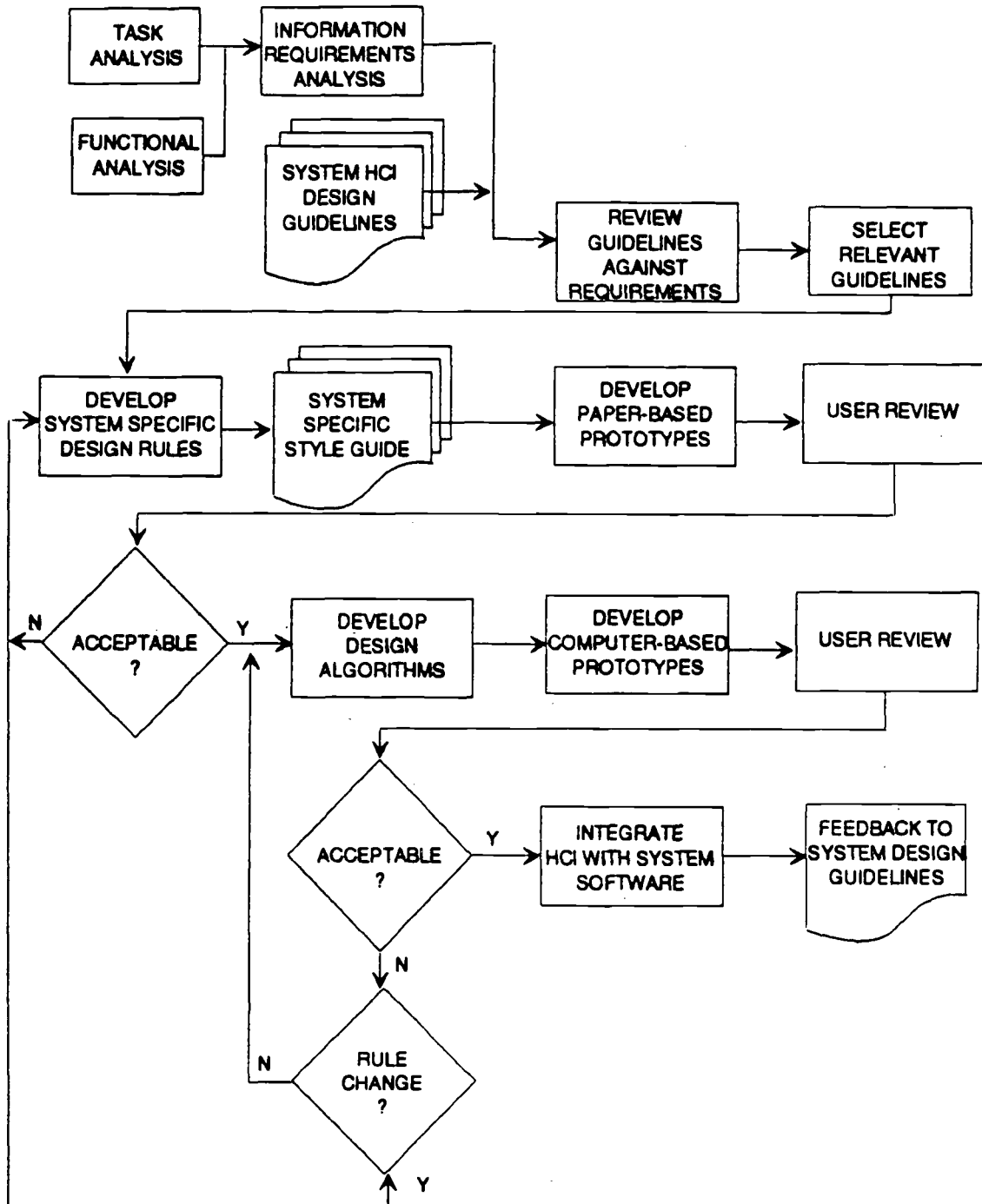


FIGURE 10.17: Sample Flowchart

- b. The layout of flowchart paths should conform to standard orientation conventions (i.e., left to right, top to bottom, or clockwise).**
- c. Consistently apply the coding schemes for flowchart elements.**
- d. Use standard directional conventions when using arrows to connect elements of flowcharts.**
- e. Use highlighting to direct a user's attention to elements of particular significance.**
- f. When using flowcharts as decision aids, require only one decision at each step, and provide the user with a logically ordered list of available options.**
- g. Use consistent wording for options displayed at decision points.**

REFERENCE LIST

Paragraphs

References

10.1.1.1	Smith and Mosier (1986) para 2.4.8-3
10.1.1.2	Smith and Mosier (1986) para 2.4.8-3
10.1.1.3	Smith and Mosier (1986) para 2.4.8-15
10.1.1.4	Smith and Mosier (1986) para 2.4.8-4 and 2.4.8-5
10.1.1.5	Smith and Mosier (1986) para 2.4.8-7 through 2.4.8-9
10.1.1.6	Smith and Mosier (1986) para 2.4.8-13
10.1.2.2a	Lewis and Fallesen (1989) para 2.5.6.1.1.2
10.1.2.2b	Smith and Mosier (1986) para 1.6-17
10.1.2.3	U.S. Department of the Army (1985b)
10.1.2.3a	Lewis and Fallesen (1989) para 2.5.6.1.1.1; U.S. Department of the Army (1985b)
10.1.2.3d	Lewis and Fallesen (1989) para 2.5.6.1.1.3
10.1.2.3e	Lewis and Fallesen (1989) para 2.5.6.1.1.4
10.1.2.3f	Smith and Mosier (1986) para 2.6-8
10.1.2.3g	Lewis and Fallesen (1989) para 2.5.6.1.1.4.6; U.S. Department of the Army (1985b)
10.1.2.3h	Lewis and Fallesen (1989) para 2.5.6.1.3
10.1.2.4	Bowser (1991) p..13
10.1.2.4a	Lewis and Fallesen (1989) para 2.5.6.1.1.2 and 2.5.6.2.5
10.1.2.4c	Smith and Mosier (1986) para 2.4.8-12
10.1.3	Bowser (1991) p..13
10.1.3.1a	Lewis and Fallesen (1989) para 2.5.6.2.3
10.1.3.1b	Smith and Mosier (1986) para 2.4.8-11
10.1.3.2e	Lewis and Fallesen (1989) para 2.5.6.2.4
10.1.3.2f	Smith and Mosier (1986) para 2.4-16
10.1.3.2g	Smith and Mosier (1986) para 2.4-17
10.1.3.3b	Smith and Mosier (1986) para 2.4.8-18
10.1.3.3e	Smith and Mosier (1986) para 2.7.3-4
10.1.3.4e	Smith and Mosier (1986) para 2.4-18
10.1.3.5	Bowser (1991) p.14
10.1.3.6	Bowser (1991) p.14
10.1.4.2	Bowser (1991) p.15
10.1.4.3	Smith and Mosier (1986) para 2.4-20
10.1.4.4b	Smith and Mosier (1986) para 1.6-5

REFERENCE LIST (Cont.)

<u>Paragraphs</u>	<u>References</u>
10.1.4.4c	Smith and Mosier (1986) para 1.6-6 and 7
10.1.4.4e	Smith and Mosier (1986) para 1.6-9
10.1.4.4f	Smith and Mosier (1986) para 1.6.10 and 11
10.1.4.4g	Smith and Mosier (1986) para 1.6-12
10.1.4.4h	Smith and Mosier (1986) para 1.6-13
10.1.4.4i	Smith and Mosier (1986) para 1.6-15 and 16
10.1.5.1	Bowser (1991) p. 14
10.1.5.2	Bowser (1991) p. 14
10.1.5.3	HFS (1988); Bowser (1991) p..14
10.1.5.4	Bowser (1991) p. 14
10.1.5.4b	Bowser (1991) p. 15
10.1.5.5	Bowser (1991) p. 15
10.2	Smith and Mosier (1986) para 2.4.6-2, 5.2
10.2.1	Lewis and Fallesen (1989) para 2.1.1-3; Brown (1989) para 5.3-8, 5.10
10.2.1	Brown (1989) p. 88
10.2.1.1	Lewis and Fallesen (1989) para 2.3.6
10.2.1.2	Bowser (1991) p. 15; Lewis and Fallesen (1989) para 2.3.9
10.2.1.4	Smith and Mosier (1986) para 2.4-5
10.2.1.6	Lewis and Fallesen (1989) para 2.3.2.2b
10.2.1.7	Lewis and Fallesen (1989) para 2.3.2.2c
10.2.1.8	Smith and Mosier (1986) para 2.4-4
10.2.2.1	Smith and Mosier (1986) paras 1.6.1-1 and 1.6.1-18
10.2.2.2	Smith and Mosier (1986) para 1.6-19
10.2.2.3	Smith and Mosier (1986) para 1.6.1-2
10.2.2.4	Smith and Mosier (1986) paras 1.6.1-2 and 1.6.1-4
10.2.2.5	Smith and Mosier (1986) paras 1.6.1-5 and 1.6.1-6
10.2.2.6a	Smith and Mosier (1986) para 1.6.2-1
10.2.2.6b	Smith and Mosier (1986) para 1.6.2-2
10.2.2.6c	Smith and Mosier (1986) para 1.6.2-3
10.2.2.6d	Smith and Mosier (1986) paras 1.6.2-4 and 1.6.2-5

REFERENCE LIST (Cont.)

<u>Paragraphs</u>	<u>References</u>
10.2.2.7	Smith and Mosier (1986) paras 1.6.2-6, 1.6.2-7 and 1.6.2-18
10.2.2.8	Smith and Mosier (1986) paras 1.6.2-8 and 1.6.2-9
10.2.2.9	Smith and Mosier (1986) para 1.6.2-11
10.2.2.10	Smith and Mosier (1986) paras 1.6.2-10, 1.6.2-12 and 1.6.2-13
10.2.2.11	Smith and Mosier (1986) para 1.6.2-14
10.2.2.12	Smith and Mosier (1986) para 1.6.2-15
10.2.2.13	Smith and Mosier (1986) para 1.6.2-17
10.2.2.14	Smith and Mosier (1986) para 1.6.2-19
10.2.3.1	Smith and Mosier (1986) para 2.4.1-1
10.2.3.2	Smith and Mosier (1986) para 2.4-12
10.2.3.3	Brown (1989) para 4.2.7
10.2.3.4	Smith and Mosier (1986) para 2.4-14; Lewis and Fallesen (1989) para 2.4.7.1
10.2.3.5	Smith and Mosier (1986) para 2.4.1-7
10.2.3.6	Smith and Mosier (1986) para 2.4.18
10.2.3.9	Lewis and Fallesen (1989) para 2.3.1.3
10.2.3.10	Lewis and Fallesen (1989) para 2.3.1.4
10.2.3.11	Lewis and Fallesen (1989) para 2.3.1.5
10.2.3.12	DoD (1989a) para 5.15.3.6.4
10.2.3.13	Lewis and Fallesen (1989) para 2.3.1.8-9
10.2.3.14	Smith and Mosier (1986) para 2.4.1-7
10.2.3.15	Lewis and Fallesen (1989) para 2.3.3
10.2.3.16	Lewis and Fallesen (1989) para 2.4-5
10.2.3.17	Lewis and Fallesen (1989) para 2.3.5
10.2.4.1	Smith and Mosier (1986) para 2.46
10.2.4.2	Smith (1986) para 2.4-7
10.2.4.3	Smith and Mosier (1986) para 2.4-8
10.2.4.4	Smith and Mosier (1986) para 2.4-9
10.2.4.5	Smith and Mosier (1986) para 2.4-10
10.2.5	Lewis and Fallesen (1989) para 2.3.2
10.2.5.1	Lewis and Fallesen (1989) para 2.3.2.1
10.2.5.2	Lewis and Fallesen (1989) para 2.3.2.1.1
10.2.5.3	Lewis and Fallesen (1989) para 2.3.2.2a
10.2.6.1a	Smith and Mosier (1986) para 2.4.3-1

REFERENCE LIST (Cont.)

Paragraphs

References

10.2.6.1b	Smith and Mosier (1986) para 2.4.3-3, 2.4.3-4
10.2.6.1c	Smith and Mosier (1986) para 2.4.3-5
10.2.6.1d	Smith and Mosier (1986) paras 2.4.3-6 and 2.4.3-7
10.2.6.1e	Smith and Mosier (1986) para 2.4.3-8
10.2.6.1f	Smith and Mosier (1986) para 2.4.3-10
10.2.6.1g	Smith and Mosier (1986) para 2.4.3-11
10.2.6.2a	Smith and Mosier (1986) para 2.4.3-12
10.2.6.2b	Smith and Mosier (1986) para 2.4.3-13
10.2.6.2c	Smith and Mosier (1986) para 2.4.3-14
10.2.6.3	Bowser (1991) p. 15; Smith and Mosier (1986) para 2.4.4-3
10.2.6.4	Smith and Mosier (1986) para 2.4.2-3
10.2.6.5	Smith and Mosier (1986) para 2.4.5-1
10.2.6.5a	Smith and Mosier (1986) paras 2.4.5-2 and 2.4.5-3
10.2.6.5b	Smith and Mosier (1986) para 2.4.5-5
10.2.7	Smith and Mosier (1986) para 2.4.6-1
10.2.7.1	Smith and Mosier (1986) para 2.4.6-6
10.2.8.1a	Smith and Mosier (1986) para 2.4.6-3
10.2.8.1b	Smith and Mosier (1986) para 2.4.6-3
10.2.8.1c	Smith and Mosier (1986) para 2.4.6-5
10.2.8.2	Smith and Mosier (1986) paras 2.4.7-1 and 2.4.7-2
10.2.8.2a	Smith and Mosier (1986) paras 2.4.7-3 and 2.4.7-4
10.2.8.2b	Smith and Mosier (1986) para 2.4.7.5
10.2.8.2c	Smith and Mosier (1986) para 2.4.7-6
10.2.8.2d	Bowser (1991) p. 16; Smith and Mosier (1986) para 2.4.7-7
10.2.8.2e	Smith and Mosier (1986) para 2.4.7-8
10.2.8.2f	Smith and Mosier (1986) paras 2.4.7-9 and 2.4.7-10
10.2.8.2g	Smith and Mosier (1986) para 2.4.7-12

11.0 APPLICATIONS

Military operational systems continue to develop automated decision aids in support of military operations. Although what constitutes a decision aid has been debated, it is important to point out that decision aids assist, rather than replace, human decision-makers. Consequently, when defining decision aids, applications limited to managing information are usually excluded, as are those that make decisions in a fully autonomous mode. The first part of this chapter (11.1) focuses on decision aids.

The question of "when to use decision aids" is reviewed in the first part of Section 11.1. Given that decision aids are to be used, the next step is to define the requirements. When the requirements are firm, the features needed to support the requirements become important. This section then deals with specific issues of decision aid interface design.

Section 11.2 initially reviews database terminology, types of database queries, and the methods used to store data in databases. Then, the section provides general guidance on user-oriented database design. The remainder provides specific guidance on query screen designs, user requirements, user-friendliness, database searching, and design requirements for novice and expert user interfaces.

Section 11.3 of the *Style Guide* includes general guidance for embedded training, followed by guidelines pertaining to more specific features. This section also contains a series of figures illustrating embedded training. Each figure is based on a basic screen prototype (see Figure 11.4). To illustrate a particular guideline clearly, a number of the figures show only a portion of the basic screen display.

11.1 DECISION AIDS

Decision aids may be designed to be parts of other software or as stand-alone applications. As an example, decision aids have been designed to assist users in evaluating military courses of action. These applications present alternatives and supporting evidence, as well as assist the user in evaluating the alternatives. The user retains a major role in developing the final recommendations. Information management software such as database management, text processing, and graphics applications may support the decision process but are not usually considered decision aids. Other applications, such as engine diagnostic software, may include many relatively

fixed rules derived from human experts. Such "expert systems" can include many properties of decision aids, but they place relatively more emphasis on internal rules to arrive at conclusion. Examples of autonomous systems include automatic fire control systems and robotic devices. These systems may require human supervision but rely heavily on internal rules and algorithms for their operation.

It is difficult to make a distinction between decision aids and expert systems (which include autonomous capabilities), since both require cooperation between human and automated system components. Holtzman (1989) differentiates between expert and intelligent decision systems and points out which is appropriate to use based on subject matter, circumstance, and preference of the decision-maker. Expert systems may have a relatively large knowledge base and rules that respond to constant environmental factors, whereas decision aids place more burden on the decision-maker. Decision aids provide assistance and are designed to help in uncertain or novel situations. The guidelines presented in this section are appropriate for both expert systems and decision aids. Therefore, the term "decision aid" or "aid" will be used to refer to both types of decision support applications.

11.1.1 WHEN TO USE DECISION AIDS

Decision aids should be used to compensate for known limitations in human decision-making and to offset the adverse effects of external factors. In general, difficulties can arise because of the fundamental limits of human cognitive (i.e., mental) abilities and lack of experience. Difficulties also arise because of various environmental factors that both stress the decision-maker and determine the type, quantity, quality, and rate of information presented.

11.1.1.1 Cognitive Considerations

Consider the cognitive limitations and styles of decision-makers when designing decision aids. For example, overload (i.e., stress, information, situational, etc.) often causes users to focus on a subset of the available information. Innate abilities and learned information-processing strategies may cause additional problems. The following points describe several commonly occurring limitations.

a. Cognitive Limitations

Novices, individuals lacking confidence, and those performing tasks under stressful conditions will make errors that may result in less-than-optimal decisions. For example:

- 1) Humans often have difficulty retrieving, retaining, representing, and manipulating large amounts of information. They also may have difficulty combining multiple cues or criteria or performing computational tasks. These difficulties result in delaying performance or avoiding difficult tasks.
- 2) Humans often have difficulty making decisions in times of uncertainty.
- 3) Novices do not have previous experience, so they often fail to recognize errors.
- 4) When making a decision, humans often simplify decision problems by selectively perceiving data, information, and knowledge. They may set outcome objectives and then look for a decision that meets them. Thus, they may focus on confirming rather than on refuting evidence. They also may adjust decision methods to fit goals or desired results.
- 5) If a decision leads to a negative result, humans may attribute the outcome to chance or to the complexity of the problem, rather than to their own decision-making deficiencies.
- 6) Humans have limited memory available for current tasks and will lose some information within seconds.
- 7) Humans have limited abilities to organize information.
- 8) Humans usually have difficulties with symbolic and quantitative manipulation of mental representations, and they may have difficulty formulating or dealing with abstractions.
- 9) Humans may have difficulty extrapolating time and space information.

- 10) Humans may fail to use prior experience to generalize in new situations.

b. Pitfalls of Complexity

Humans often have difficulty dealing with complexity and may, therefore, try to make a problem less complex by avoiding certain aspects of it. Consequently, they may not consider all factors when making decisions. Some strategies humans use to make decisions less complex are:

- 1) Humans often simplify decision problems by only considering a few alternatives.
- 2) Humans may use only part of the available information. Or they may use information that corresponds to a mental representation or model of what they imagine the solution to be, even if this means rejecting or misperceiving relevant information. They may combine or "chunk" information in various ways, rely on poor memory-search strategies, or rely on erroneously perceived correlations between data.

c. Cognitive Biases

Humans have biases that can carry over into the decision-making process.

- 1) Humans may recall information that has been recently acquired, frequently rehearsed, or semantically related to current information.
- 2) Humans may anchor their judgments (i.e., place greater emphasis on early evidence) and then fail to adjust when provided new information.
- 3) Humans may give preference to information they believe is causally related to the problem.
- 4) Humans may provide numeric judgments that contain systematic bias or variance.

- 5) Humans may select cues that are often unreliable indicators of the true situation.
- 6) Humans may use inappropriate analogies to generate and compare options.
- 7) Humans may incorrectly identify current situations with similar past events.
- 8) Humans may fail to detect unique features among similar cases, and inconsistent or ambiguous information may not be noticed or emphasized appropriately.

d. Time Allocation

Humans may fail to allocate time properly to different phases of the planning process. Too much attention to early stages of planning may leave inadequate time to evaluate derived alternatives properly.

- 1) Humans may perform detailed analysis early but fail to do so later.
- 2) Humans may fail to develop and evaluate the alternatives thoroughly.
- 3) Humans may fail to identify, evaluate, compare, and combine salient information and, therefore, fail to identify, prioritize, and assess goals.
- 4) Humans may fail to model (war-game) alternatives because of lack of time.

11.1.1.2 External Factors

Many external factors may influence the quality of decisions.

a. Information Overload

When the complexity, dynamics, and/or volume of information to assess are high (such as in battlefield operations), they may degrade decision-making performance.

b. Time Stress

Humans have difficulty analyzing information fast enough to meet external time constraints. When they have enough time, they often have difficulty maintaining high performance long enough to analyze all data.

c. Limited Information

Decision-makers may work in situations where information available to support their decisions is limited. Under such circumstances, lack of experience and human limitations in making or formulating estimates may pose problems.

d. Training

Decision-makers may not have the experience, deductive skills, or knowledge of the procedures necessary to make a decision. A decision aid can assist by performing some steps, by leading the human through the required steps, and by filling in knowledge gaps. Properly designed decision aids also train users through explanation and embedded training.

11.1.1.4 When to Use Decision Aids

Use decision aids to help the user overcome the difficulties previously described. For example:

a. Manage Complexity

Decision aids help the user cope with information overload; they focus attention. Use decision aids when the user is trying to manipulate large amounts of data or visual representations, combining multiple criteria, allocating resources, managing detailed information, and selecting and deciding among alternatives.

b. Improve Timeliness

A decision aid helps a user perform many time-consuming activities more quickly. Some examples include diagnosing the current state of a system and mathematical calculations, particularly when they are beyond the user's abilities. Providing aid when users encounter unfamiliar problems also helps improve the timeliness of the process.

c. Best Use of Limited Data

Use decision aids when limited data result in uncertainty. Decision aids help by predicting future events from limited information, improving the accuracy and reliability of critical tasks, and addressing critical areas beyond the ability of the user.

d. Overcoming Limitations

Use decision aids to overcome the human cognitive limitations described in the earlier sections. For example:

- 1) Use decision aids to overcome human limitations in dealing with uncertainty.
- 2) Decision aids are helpful in overcoming emotional components of decision-making.
- 3) If the quality of human performance is in question, decision aids can add greater accuracy to the process.
- 4) Decision aids are ideal in cases where memory- and information-retention problems exist.
- 5) Decision aids overcome cognitive biases well.

11.1.1.5 When to Consider Alternatives

The following are circumstances under which the use of decision aids may not be advisable:

a. Obvious Solutions

Do not use decision aids when solutions are obvious or when one alternative clearly dominates all other options.

b. Time Requirements

Use decision aids only when sufficient time is available or when the user is authorized to make decisions.

c. Generalizing

As appropriate, defer to the human ability to generalize.

d. Adaptation

Recognize situations where individuals may be superior in adapting to novel situations.

11.1.1.6 Cautions and Limitations

Exercise caution when introducing decision aids, in particular, when they include functions that reduce the role of human judgement.

a. User Complacency

The decision aid may encourage users to take a less active role. This, in turn, may cause users to be inattentive and less prepared to handle sudden decreases or increases in workload, both of which may reduce accuracy.

b. Continued Vigilance

If the user's role becomes less active, the user may have difficulty maintaining sustained attention, which may lead to longer user response times.

c. Discrimination Limitations

It is necessary to recognize limitations in a user's ability to discriminate between correct and incorrect automated decisions.

d. Fear of Automation

Many users mistrust automation and automated decisions, preferring to believe their performance is superior to that of automated systems. User attitudes toward automation are often based on the fear of being

replaced. The designer should take this into account when planning the role and degree of authority the user will have in overriding automated decisions.

11.1.2 DEFINING DECISION AID REQUIREMENTS

Develop decision aids or expert systems that focus on tasks that users find difficult, rather than on what is already done routinely.

11.1.2.1 Understand Tasks

Base designs on an in-depth understanding of both the tasks to be performed and the conditions of their performance.

- a. The best way to define decision aid requirements is to start with experts in the field. However, it is important to choose the experts appropriately. Be sure to use more than one expert and verify that they really are experts. If they have knowledge of part of the field, be sure to consider those parts, and find other experts for the other parts. Also, ensure that common users participate with the group of experts. When obtaining information from the experts, provide a means to identify the criteria used to reach decisions.
- b. Decision aids must be matched to operational conditions and limitations they are designed to support.
- c. Recognize that not all functions are appropriate for decision aids. Determine the appropriate functions and design them to be compatible with the user's decision processes.
- d. Provide no more than one aid for each task.

11.1.2.2 Requirements

Decision aid development should be driven by requirements, not by technology.

- a. Identify areas where users actually need help, then match the decision aid to the needs of the intended users.
- b. Recognize the user's decision situation and goals, and focus on the highest-level goal.

- c. **Anticipate skepticism concerning automated decision support. Recognize that the dominant factor in accepting decision aids is perceived utility. The system must add new capabilities or increase efficiency in the performance of decision-making tasks.**
- d. **Consider characteristics of the user population in designing the decision aid and its interface.**

11.1.2.3 Types of Aids

Types of aids and presentation formats may vary according to the phases of the decision process (i.e., alerting, acquisition, evaluation, and responding) and factors such as time stress.

11.1.2.4 Function Allocation Between Humans and Computers

Allocating functions between humans and computers must be based on cognitive task analysis, not on what is achievable using current technology.

- a. **Recognize that aided performance may not exceed unaided performance, even though aided methods are preferred.**
- b. **Decision aids and expert systems can enhance decision quality. However, they may increase the user's workload because users may be required to consider more variables. Seek design alternatives that prevent or minimize increased workload.**
- c. **The aid must be complete for its intended purpose. Address all critical aspects of the decision situation.**
- d. **Recognize that the user's decision-making behavior is contingent upon the task and the context within which it is performed. Design the decision aid to provide decision methods suitable to probable variations in tasks and context.**
- e. **Users often prefer to perform some of the tasks and allow the decision aids to perform others. Specifically, users prefer to do the easy to moderately difficult tasks and leave difficult tasks to the decision aid. This interaction is necessary to maintain user interest and attention. Decision aids are more acceptable to users if viewed as advisors rather than decision-makers.**
- f. **Avoid applications that are trivial or lack complexity because they may**

undermine the value of automated decision support methods.

11.1.3 FEATURES OF DECISION AIDS

11.1.3.1 General Design Considerations

- a. Ensure that a decision aid is easier to use than the decision process it replaces. It must be flexible, versatile, and easy enough to benefit typical users (i.e., users don't need to be subject matter experts). A decision aid must use terminology and criteria appropriate to the target user group. It must be easy to control and understand.
- b. Ensure that a decision aid is capable of responding to the user's ad hoc requests in time to allow the information to influence decisions. The interface should facilitate the exchange of information.
- c. Tailor decision aids to the resources available to the user.
- d. Ensure that a decision aid automatically identifies meaningful patterns and relationships and brings them to the attention of the user.

11.1.3.2 Provide Decision Alternatives

- a. Ensure that a decision aids is able to support development and evaluation of multiple, feasible alternatives. The aid should present a set of possible alternatives, each of which could be feasible. However, the aid should not display all of the options when that would be too complex. The decision aid also should display which goals are served by the different alternatives and applicable options.
- b. Ensure that the decision aid supports user evaluation of decision options. First, the aid should generate alternatives for the user to evaluate and should allow the user to input his or her own alternative(s). Second, the aid should have a method of assigning and explaining probabilities for alternatives. The user should be able to explore different solutions, including using different decision strategies and criteria. Once the user has applied all desired options, the aid should rank-order the decision alternatives. This assistance also should include guidance in using rating procedures.

11.1.3.3 Prediction, Simulation, and Modeling

- a. Ensure that the application is able to predict future data. Historical data should be available to make comparisons, search for precedents, and assist the user in visualizing trends. The decision aid should alert the user when it predicts a future problem or opportunity upon which the user needs to act.
- b. Provide a modeling and simulation capability to support "what if?" exercises and to make predictions based on current conditions.
- c. Ensure that models used in decision aiding are appropriate and validated.

11.1.3.4 Identify and Assess Factors Underlying Decisions

- a. Provide a means of obtaining and assessing weights for multiple criteria. Multiple criteria should be statistically independent, when possible. As appropriate, provide a means of combining weights from multiple sources. This refers to the technique of multi-attribute decision-making.
- b. Identify and rank causal factors by their importance, and assign weights. The application should allow users to modify the decision factors and their weights and to provide and adjust risk factors used in decision models. This refers to techniques such as pair-wise comparison.
- c. Ensure that the aid is able to explain the contributions of underlying factors and supports the use of sensitivity analysis for exploring those contributions. The aid must identify and assess operational constraints and provide a means of informing the user (upon request) of decision-aid boundaries or other limitations. The aid should make available to the user the assumptions underlying modes and parameters and a history of the aid's past performance.
- d. Ensure that the decision aid makes it easy for the user to provide input into the aid's decision. The user should be able to add new decision factors and set the range of conditions (within the decision aid's set limits), the level of output detail, and the parameters for optimization. Provide a means for saving and reusing the user's modifications, but also provide a means to return to the default settings.
- e. Assist in visualizing interacting factors.

- f. Provide a means for assuring the validity of elements added to the decision model, in particular those used over successive applications.

11.1.3.5 Handling Decision Aid Recommendations

- a. Ensure that the application is able to calculate and display results of selected decision options.
- b. Ensure that the application provides facilities for assessing costs, risks, and benefits of all alternatives.
- c. For users to trust the decision aid, the aid must explain the rationale behind outputs or recommendations. The aid also should provide indicators of certainty or uncertainty when making recommendations.
- d. When data are missing or uncertain, ensure that the aid identifies this situation and gives information on the possible impact on the recommendations.
- e. Ensure that the decision aid includes internal consistency checks to prevent the system from making contradictory predictions and recommendations.
- f. Ensure that the decision aid informs the user when it cannot handle the current situation.

11.1.4 USER REQUIREMENTS

11.1.4.1 General Considerations

- a. Ensure that the decision aid is user-friendly, beneficial to the user, and presents information that is readily understood by or familiar to the users. Where possible and appropriate, ensure that the decision aid has sufficient "intelligence" to adjust to user task requirements.
- b. Ensure that the decision aid uses decision methods acceptable to the decision-maker and is able to accommodate user changes. Once the decision method is determined, the user must retain control throughout the process. The aid should provide feedback on the method and the current stage of processing.

- c. **Reduce the user's data-entry requirements as much as possible. To do so, set defaults for data-entry fields. However, these defaults and fields must be user-changeable.**
- d. **Ensure that a decision aid automatically alerts users to important new developments occurring in the database or as a result of predictive modeling.**
- e. **Ensure that the system encourages the user to participate in the decision process. To do this, the system should represent problems and solutions in the same way the users do. The system also should try to foster user "ownership" of decisions and allow the user to exercise judgment over the decision aid results. This includes providing sufficient information to the user both about the process and about the end result.**
- f. **Ensure that the decision aid guides the user through the process. It should provide automated guidance on how to define and analyze a problem and formulate a decision. When user input is required, the decision aid should aid in making this requirement clear. However, it should not make the user dependent, such that the process cannot be completed when the system is unavailable.**
- g. **Avoid presenting too much data. Use aids to reduce, filter, and preprocess data into a form useful to the decision-maker.**
- h. **Avoid increasing the user's work load, when possible. Prepare users for changes and possible increases in work effort when necessary, and point out the aid's abilities to increase effectiveness.**
- i. **Reduce complexity. A major reason for using decision aids is to simplify the user's task. Therefore, some guidelines are necessary on the amount of information to present to the user. In general, the system should provide information required to perform the tasks allocated to the user; however, it should only present information relevant to the task being performed. The system should provide no more information than is essential and should avoid repeating already available information. Present the information using a level of abstraction, resolution, or detail appropriate to the immediate task.**
- j. **When time is limited, ensure that the system anticipates the user's needs and provides a greater degree of autonomous decision-making.**

- k. Ensure that users are able to extend and personalize the decision aid. However, provide a means to validate models created or modified by users. Provide sufficient warnings about the consequences of failures to validate.
- l. Ensure that the decision aid's analysis is flexible to the user's needs and desires and give the user control over the data retrieval and analysis process. The user should select the degree of analysis to be done and the time frame to be considered. When the system asks questions, the user should have the option of either changing the question or not answering. The system should also accommodate the different information requirements of commanders and staff users, including the ability to adjust the level of detail. It should be able to create a user profile containing preferences and jargon.
- m. Provide procedures appropriate to the user's level of expertise. Designers should recognize that experts may use mental imagery; novices depend more on rule-based procedures.

11.1.4.2 Decision-Aid Interface

- a. Ensure that the user-machine interface supports an intelligent dialogue between the user and the decision aid. For example, it should adapt to the user; understand the user's goals, needs, and abilities; interpret poorly formulated queries; correct user errors; and overcome user limitations. The interface also should reflect the tasks to be performed and should be tailored to the resources available.
- b. Ensure that the system helps prevent the user from making errors. When errors are made, it should provide automatic error recovery.
- c. Apply user-interface design guidelines mentioned elsewhere in this document.
- d. Ensure that the decision aid allows users to customize formats to their own needs. However, it is preferable to minimize the user's requirements to make such changes. To do this, the application should associate and group data in a meaningful way, and displays should match the task.
- e. See Section 8.2 of this Style Guide for detailed information on HELP applications.

11.1.4.3 Explanations

The decision aid should be capable of providing domain-specific explanations to answer user questions. Decision aids must be capable of guiding the user through the decision process, as well as providing procedural help on use of the system.

- a. When the system provides explanations, ensure they are easy for the user to understand. Explanations should use terms familiar to the user, incorporating the user's concept of the problem and maintaining consistency with the immediate task. Intuitive explanations or analogies are helpful for topics that are likely to be too difficult for the user to understand.
- b. Length of explanation is important. Provide a short explanation initially, with the ability to provide more detail at the user's request. Consider how much to tell the user. Weigh trade-offs in what the user can learn about the decision aid and what the decision aid can/should explain to the user.
- c. Assist the user in locating key elements of the decision model, as related to a specific decision task.
- d. Provide the capability to explain the current decision model or method, and be prepared to justify the use of component factors. Document the decision aid's algorithms, and make them available for user inspection.

11.1.4.4 Training

- a. Provide backup systems and appropriate training in performing any user tasks replaced by decision aids. When decision aids are available, provide regular training to the user in all skills required to maintain proficiency on backup systems. This training will be necessary if decision aids become unavailable. Training may be preferable to using decision aids for handling infrequent critical events occurring in dynamic environments.
- b. Train users to recognize inappropriate uses of the aid and to recognize errors. Provide readily accessible lists of limitations; include information concerning limitations and errors in embedded training. Users should learn not to categorically accept a decision aid's capabilities.

11.1.4.5 Decision Graphics and Displays

- a. Prepare graphics, textual reports, and input screens in formats familiar to the user. This will facilitate rapid and accurate information-processing. However, the user should be able to control formats or to select from alternate preprogrammed formats.
- b. Graphics are another important part of the user interface. Graphics help assist the user in visualizing information. However, guard against inaccurate graphics, as they can have a strong negative impact.
- c. Provide historical displays of comparative cases, to include time-sequenced presentations.
- d. Use spatial rather than textual formats when the task involves extensive spatial processing, in particular when task performance time is limited. Use tables rather than graphs when reading specific data points.

11.1.5 ORGANIZATIONAL FACTORS

11.1.5.1 Information Requirements

Ensure that decision aids are flexible in meeting the different information requirements at different echelons.

- a. Different levels of command require different levels of abstraction. Ensure that decision aids accommodate different levels of detail and time constraints at each echelon.
- b. Ensure that command and control decision aids are distributed (i.e., they should support multiple, cooperating decision-makers at different locations sharing a common database).
- c. Where practical, design decision aids to support the entire command and control process, rather than develop them to support isolated phases of the process.

11.1.5.2 Entire Organization

Ensure that decision aid designs consider impacts on the entire organization, particularly where organizational goals may supersede those of subordinate decision aid users.

11.1.5.3 Complementary

Ensure that decision aids complement existing tasks and information-distribution systems.

11.1.6 FLEXIBILITY

11.1.6.1 Change-Over Time

Design decision aids as adaptive systems (i.e., they must accommodate growth and evolve over time to meet changing conditions, doctrine, etc.).

- a. Establish policies for implementing changes, as well as the mechanisms for those changes.
- b. Adjust to changing situations and user preferences (different circumstances and users may require different methods).

11.1.6.2 Maintainability

Ensure that decision aids are maintainable by the user. Rules, data, and decision logic should reflect current needs.

11.1.6.3 Type of Support

Allow the user to tailor the type of support provided by the decision aid in the presence of changing conditions.

11.2 QUERY LANGUAGE

A database consists of interrelated data that are searchable by a computer. The computer software that facilitates processing information into organized or summarized groups is called a Database Management System (DBMS). Retrieving information from a database involves identifying a set of items that match or are similar to the user's query or statement of information need. The term "data access" refers to the process of locating and retrieving requested sets of data. By contrast, the term "data presentation" refers to the process of displaying that data to the user in an appropriate fashion. Data access is the query, and data presentation is the result.

The software that makes up the user interface usually consists of applications

programs, report program generators, and query languages. The applications programs allow the end users to enter, retrieve, and update the data in the database. Report-generator utility programs help users specify the content and format of reports. Query languages are used to meet requests for information or to provide a means to browse through the database.

Databases must usually be searched in a series of steps. The computer-readable message containing the search terms and logical operators for combining them must be derived from the search query or queries submitted by the user. The search terms are then matched against terms in the database file, either indirectly by searching the index or by directly searching records. The computer responds with counts of retrieved items and should allow the user to sample the items by displaying them on the screen. The user can then make iterative adjustments, either to broaden or narrow the scope of the query.

TYPES OF QUERIES

Users most often communicate with databases by means of command-driven (i.e., query languages), form and menu-driven, natural-language, and icon-based interfaces.

Command languages provide flexibility and relieve the experienced user of the requirement to traverse an entire menu structure to select a command. Users of this type of interface must be familiar with the command language, the steps required for solving problems, and the computer's syntax for accomplishing each step of the process. Structured Query Language (SQL) and Query-by-Example (QBE) are commonly used languages that perform similar functions.

SQL is a textual language that is becoming a relational database standard. SQL includes table definition, database update, view definition, and privilege-granting, in addition to query facilities. SQL is often embedded in programs written in other languages, where it generates query results that can be processed by programs written in the host language.

QBE is a table-oriented version of the SQL relational database language and is often supported where SQL is used. QBE provides a pictorial representation of database tables. Symbols placed in the proper table columns specify query selection conditions, grouping, data display, and database updates. Although QBE's tabular format offers advantages to users, it requires user sophistication for effective use.

Another screen-oriented method provides a relatively easy technique for developing queries. Query By Forms (QBF) presents the user with data-entry forms that also can be used as templates when developing queries. When accessing data, the user can select one or more of the data-entry fields and enter values, ranges of values, or logical conditions, which are then automatically translated to database queries. Using familiar forms for data entry and search tasks facilitates the user's performance in creating straightforward queries.

Menus provide user-friendly interfaces to command languages, such as SQL. They are designed in a hierarchical or tree structure, which allows the user to proceed step by step through the menu structure to the desired level of detail. Some menu systems allow the user to go directly to a specified level by keyboard command or selecting items from a multi-level menu map. Menu-based query aids offer several advantages. They lead the user through the problem-solving process by indicating which options are available at each point. They are comparatively easy for a novice to use, particularly when the user is not familiar with the query command structure (low syntactic knowledge) or is uncertain how to proceed in solving a particular problem (low semantic knowledge). Menu systems also have disadvantages. Users are forced to make selections from the choices offered by the system and are, therefore, subject to any constraints that might be present. If a user makes an incorrect choice at any level, it can be time-consuming and frustrating to retrace the steps in the menu structure.

Natural language interfaces allow users to formulate queries in their native language (e.g., English, Spanish, etc.). These interfaces use a knowledge of syntax (grammar) and semantics (meaning) to interpret queries and translate them into the query language used by the database system. This approach frees the user from learning the usual conventions and rules of query language. Although natural language interfaces offer great potential, they may require considerable user effort in setting up the underlying dictionaries.

Users may directly query icons, maps, schematics and other visual depictions of physical objects by using a pointing device to select the picture or its features in some sequence. Pointing devices (e.g., a mouse, touch-sensitive screen, or trackball) are often used in combination with menus and text-entry

screens to formulate queries. Direct interaction with visual representations of physical objects and icons can facilitate human performance.

DATABASE DESIGN

Ease of use and overall performance of a database system depend on its file structure (the manner in which the records are organized in the file or database) and search processes. The details are chosen by the designer or programmer of the system, often with more concern for the programming aspects of a particular model than for the human performance constraints imposed by that model. The optimum form of information representation will be a function of the task being performed. Unfortunately, current research offers little guidance on how to proceed in database retrieval situations.

Database designs typically use hierarchical, network, relational, or object-oriented models. The hierarchical model represents data in tree structures, and networks represent data as interconnected structures of records linked in one-to-one or one-to-many relationships. Relational databases organize data in tables. Because of its power and ease of use, the relational representation is the prevalent model today and is likely to be the database model of choice in the near future. Object-oriented (sometimes called extended relational) database systems are considered to be part of the next generation of database systems. An object-oriented system represents real-world entities as "objects" that have attributes and defined relationships with other objects.

It is important to recognize that each of these database models can influence the format in which information is presented and the way in which the user can add to, retrieve, or change the information contained in the database. In the end, database models determine the modes of user-database interaction, the format in which the data are presented to the user, and the ease with which a user can acquire information from the database:

11.2.1 GENERAL RECOMMENDATIONS

11.2.1.1 Ease of Use

Ensure that a query language or procedure is easy to learn and use. In fact, ease of use and user-friendliness often determine whether the database is used. A program will not be used if it is intimidating, is too difficult, or requires too much effort.

11.2.1.2 Interactive Queries

Give preference to on-line query over batch or off-line modes because it provides the user the opportunity to interact with the system.

11.2.1.3 User Assistance

Ensure that an application assists the user in creating complex queries and in narrowing down the search in a step-by-step fashion.

11.2.1.4 Error Detection

Alert the user to syntax errors in queries and, if possible, to semantic faults (semantic integrity).

11.2.1.5 Minimum Training

Require only the minimum training. An effective user interface should not require extensive training to be used easily.

11.2.1.6 User-Oriented Designs

Design the system interface in cooperation with the end users to ensure their satisfaction with the final product. User involvement is most effective when users participate in both development and implementation of the system interface.

11.2.1.7 Multiple Search Options

Consider the nature of the searches to be performed before choosing an interface format. When more than one type of query is possible, one solution is to choose the interface format that provides the best average performance. Alternatively, provide multiple query and display formats, so the user can change formats as desired or when the nature of the search task changes.

11.2.1.8 Appropriate Displays

Ensure that displays are appropriate. The forms of information display that facilitate quick responses are not necessarily the same forms that produce accurate responses. The three basic forms of information display are spatial, verbal, and tabular formats. Pictures (spatial) are superior to words (verbal) in recall and recognition tasks and often lead to quicker completion times on procedural tasks. However, words lead to greater accuracy in performance.

11.2.1.9 Individual Preferences

Individual preferences play an important role in the effectiveness of any query application. Users perform better and provide a higher proportion of correct answers when the format of the database matches the format they prefer. Observation shows that, although experience with an application can lead to changes in preference, only pre-existing preferences for display formats influenced user performance.

11.2.1.10 Displaying Results

Display data numerically or graphically. Graphical displays include the bar graph, plot, pie chart and other computer-drawn pictures. Because graphical presentations provide less accuracy than numerical presentations, the most important consideration is the transfer of meaning to the user. See Figure 11.1.

11.2.2 QUERY SCREEN DESIGN

Query screens display the results of a query request or the contents of computer files. The objective in query screen design is to aid the user in

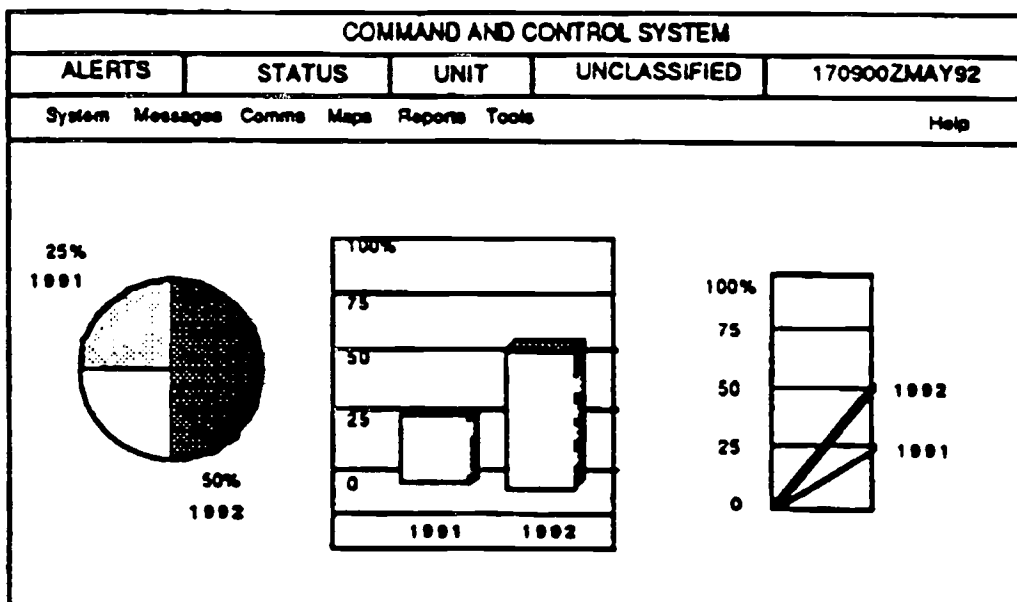


FIGURE 11.1: Examples of Pie Chart, Bar Graph, and Line Graph

quickly and easily locating data or information. Query screen development should optimize human scanning, as scanning is easier when eye movements are minimized, required eye movement direction is obvious, and a consistent pattern is followed.

11.2.2.1 Screen Design Principles

- a. Include on a query screen only information that is relevant to that screen. Forcing a user to wade through volumes of data is time-consuming, costly, and error-prone. If information will never or very seldom be used, do not display it. An item may be relevant one time a screen is displayed but irrelevant another. Limit a transaction or screen to whatever is necessary to perform actions, make decisions, or answer questions.
- b. Ensure that the interface display groups information in a logical or orderly manner. Locate the most frequently requested information in the upper left corner.
- c. Locate the most frequently requested information on the initial screens for multiscreen transactions.
- d. Ensure the screen is not overloaded, and use spaces and lines to balance the screen perceptually.
- e. Use consistent terminology, commands, formats, and general appearance throughout the interface. Ensure learning can be transferred between modules of the program.

11.2.2.2 Query Screen Organization

Organize the query screen in a logical, orderly, and meaningful manner. When information is structured consistently with a person's organizational view of a topic, that person comprehends more information. Finding information on a query screen can be accelerated by many factors, including the following.

- a. The interface should locate the most frequently sought information on a screen in the upper left-hand corner. If there are multiple screen transactions, locate the most frequently sought information on the earliest transaction screens.
- b. To aid the user in locating a particular item, provide easily scanned and identifiable data fields. Accomplish this through columnization with a top-to-bottom, left-to-right orientation, which permits the eye to move easily left

to right across the top of the columns to the proper column before beginning the vertical scan.

- c. Top-to-bottom scanning will minimize eye movements through the screen and enable human perceptual powers to be used to the fullest.
- d. Current technology presents query output mainly in tabular format. Emerging object-oriented technology will provide different ways to present such information visually.

11.2.2.3 Captions (Labels)

- a. Captions should be complete and written in clearly understandable language. Display captions in upper case, although lower case may be used for long, descriptive captions. Do not use reverse video or highlighting for labels.
- b. For single fields, locate the caption to the left of the entry fields. Separate the caption from the entry field using a unique symbol and one blank space (a colon ":" is recommended). With multiple occurrence fields, locate the caption one line above and centered over the column of data fields.

11.2.2.4 Data Fields

- a. The application interface should provide visual emphasis to the data fields.
- b. The interface should display directly usable information, as well as fully spell out codes and compressed information. The data displayed should include natural splits or predefined breaks.
- c. The interface should display data strings of five or more characters (numbers or alphanumeric), with no natural breaks, in groups of three or four characters with a blank between each group. Data strings should be left-justified, and numeric data should be right-justified or justified about the decimal point. For all types of data, identical data should be consistent despite their origin. See Figure 11.2.

COMMAND AND CONTROL SYSTEM					
ALERTS	STATUS	UNIT	UNCLASSIFIED	170900ZMAY92	
System	Messages	Comm	Maps	Reports	Tools
					Help
Examples:					
String		SS450285184			
Display		SS450 28 5184			
Justification		Name: John Doe			
Alpha		Phone: 555-2000			
Numeric		\$ 922			
		1,044			
		21			

FIGURE 11.2: Data Layout and Justification

11.2.2.5 Data Organization

- Organize data in accepted and recognizable order, with vertically aligned captions and data fields in columns.
- Ensure the application justifies data displays consistently.
- Promote readability by designing the interface with at least one space between the longest caption and the data field column, and with at least one space between each heading. Section headings should be on-line above related screen fields, with captions indented a minimum of five spaces from the beginning of the heading and fully spelled out.
- When presenting multiscreen transactions, place a screen identifier or page number in the upper right-hand corner of the display (i.e., "screen 2 of 5").
- Locate error and status messages consistently in a separate area of the screen. Emphasize these messages by using a contrasting display feature (e.g., reverse video, highlighting, or preceding series of unique symbols, such as asterisks).

- f. Provide different forms of information display for different search tasks. For example, the interface should provide a selection of display formats as well as a review format where certain fields of the retrieval records can be reviewed without retrieving the entire record.

11.2.3 USER REQUIREMENTS

11.2.3.1 Search Enhancements

- a. Query optimizers are software procedures that automatically enhance the ability of the database application to execute queries. Use query optimizers to increase the effectiveness of the program, but they should be invisible to the user.
- b. Permit the user to rank search terms by importance. Then use this ranking in a formula for automatically ranking records by relevance in the retrieval set.
- c. Provide additional search terms in a retrieval set. For example, use a memo field to list the additional search terms related to a particular field.
- d. Ensure that the application allows redisplay of results of the previous search without requiring reprocessing.

11.2.3.2 Automatic Functions

- a. Provide automatic recognition of spelling variants (e.g., color versus colour).
- b. Provide automatic recognition of acronyms.
- c. Provide automatic recognition of variations in romanization (e.g., Peking versus Beijing).
- d. Provide automatic inclusion of the inverted form (e.g., Newborn Infant to Infant, Newborn).
- e. Ensure that the application automatically removes punctuation from search terms when matching them against search-key values.

11.2.3.3 Word Stemming

- a. Ensure that the application uses a set of rules for reducing words to their root forms by stripping them of their suffixes (e.g., reduce, reduction, reducing).

- b. Ensure that the application automatically searches the index for all words containing a given root (e.g., the word "form" is the root of formation, inform, and information).
- c. Provide rules for exceptions based on the language of the discipline or specialty area.
- d. Allow truncation. The application should automatically search for all words or phrases that begin with the same character stem (e.g., term for terms, termination, and terminated).

11.2.3.4 Erasing

- a. Allow immediate deletion of individual characters or deletion of the entire line of input (provided it has not been processed by the computer).
- b. Permit deliberate interruption of computer messages or displays without disconnection (break or interrupt key).

11.2.3.5 User Satisfaction

User satisfaction with the system can be enhanced by including those factors described in the following paragraphs.

- a. Provide results in a timely manner. One factor of timeliness is the elapsed time from when the command is sent until a response is displayed (response time). Another is the time required for characters or graphics to appear on the screen or hard-copy device (display rate).
- b. Ensure the appearance, print format, and organization of output are natural to the user. User-generated report formats aid in matching the appearance of the output to what the user expects.
- c. Minimize the level of effort required by the user, including the limitations or qualifications that the application places on search output.
- d. Provide maximum capability to the search system while maintaining maximum retrieval effectiveness.
- e. Ensure that the application assists the user in formulating searches for maximum usefulness of the search results.

11.2.4 USER-FRIENDLINESS

11.2.4.1 Commands

- a. Use mnemonics to avoid the need for remembering syntax (i.e., as sequences or specifications in output instructions).
- b. Use commands in an easy-to-learn, user-oriented system language.
- c. Use unambiguous commands. The meaning should be clear to the user.
- d. Ensure that entering data is not physically awkward for the user, and keystrokes are limited to those absolutely necessary. Provide the capability to define Ctrl key, Alt key, or function key combinations (i.e., Ctrl/Alt/Del to reboot the system) in place of keystroke combinations.
- e. When a command will delete stored information, provide a complementary command that reverses the action. If deletion is irreversible, provide the user with the opportunity to reconsider the action. The application should check for meaningless commands against a list of authorized commands, after which the application should allow the user to enter a revised command rather than automatically abort the procedure.
- f. Provide the user with abort or escape facilities for controlling the dialog flow.

11.2.4.2 Computer Messages

Messages should be clear, simple, and concise. Present the user with the briefest message that can be properly interpreted. Directive messages should be specific and in the context of the current working environment. Messages should warn the user of irreversible action.

11.2.4.3 Error Messages

Deal with mistakes in a positive, helpful manner. Users will thus gain confidence in the system and feel less intimidated or fearful of damaging it or the data. The error message should appear when the user enters a command that is misspelled, improperly formatted, or cannot be processed because it is inappropriate to the situation. The message should provide instructions for revising the erroneous command.

11.2.4.4 Documentation

Full system documentation should be available in manual form.

11.2.4.5 Tailor the Interface

Tailor the interface to suit the needs of users.

- a. Tailor frequently used queries. In cases where the value of only one or two parameters changes, provide the user with default values for those parameters. For example, a query might request the names of all Army officers with over ten years service who graduated from an academy in the top 10 percent of their class and who serve in the infantry. This query contains elements that could be requested several different times, using slightly different conditions each time.
- b. Macro definition procedures (user-defined commands) are an important feature for expert users who prefer to define their own commands and personalize their environments by encapsulating frequently used query sequences in a new command. Macros greatly simplify user interactions with the application as well as save time. The application should allow the user to store these macros as files or define function key combinations to perform the function.

11.2.4.6 Accelerators

Ensure that the interface provided accelerators to save keystrokes. For example, special keys can be dedicated to commonly used functions. The application should permit direct commands as alternatives to menu options.

11.2.4.7 Backup

Ensure that the application shields the user from system failure. Provide backup facilities both internally by the software application program and externally by the operating system.

11.2.4.8 Restore

Ensure that the application provides a restore utility to facilitate recovery of damaged or destroyed data from backup copies.

11.2.4.9 Interrupt

Ensure that the system provides the capability to interrupt work with the application software, then comes back later to resume work at the same point.

11.2.5 SEARCHING

11.2.5.1 Commands

Make the following types of database utility and search commands available to the user:

- a. Provide a database SELECT command.
- b. Provide commands to create and erase sets.
- c. Allow users to combine two or more sets to create new sets.
- d. Provide the capability for users to specify report formats. The user should be able to name the report, identify the relations from which the report data will be derived, determine the report layout, and define the lines and headings or captions of the report. The user should be able to save the created formulating query and report format for later use.
- e. Provide the capability for users to restrict the output of retrieval sets.
- f. Provide the capability for users to save search results easily.
- g. Provide the user a list of previous search commands upon request. The number of saved commands could be set by the user or could be a prespecified number.

11.2.5.2 Control Functions

Ensure that the application provides control functions to aid the user in dealing with the system. These functions should include signaling about the system's current state or performing an action based on the state.

- a. The input parameter for the MARK command should be the current field value, and the application should note the marked value for future reference. For example, fields or records could be marked for deletion.

- b. DESCRIBE should use as its parameter the current field value. Provide the user with a detailed explanation or description.
- c. The parameter for the DROP command should be the current field value. The current field should be dropped from the structure.
- d. The application should provide the user status information upon request. This should include the completion and success or failure of the last search operation executed.

11.2.5.3 Editing Commands

Editing commands are necessary during query formulation. The application should provide a text-editing box to be used for typing search queries. The following functions should be available. See Figure 11.3.

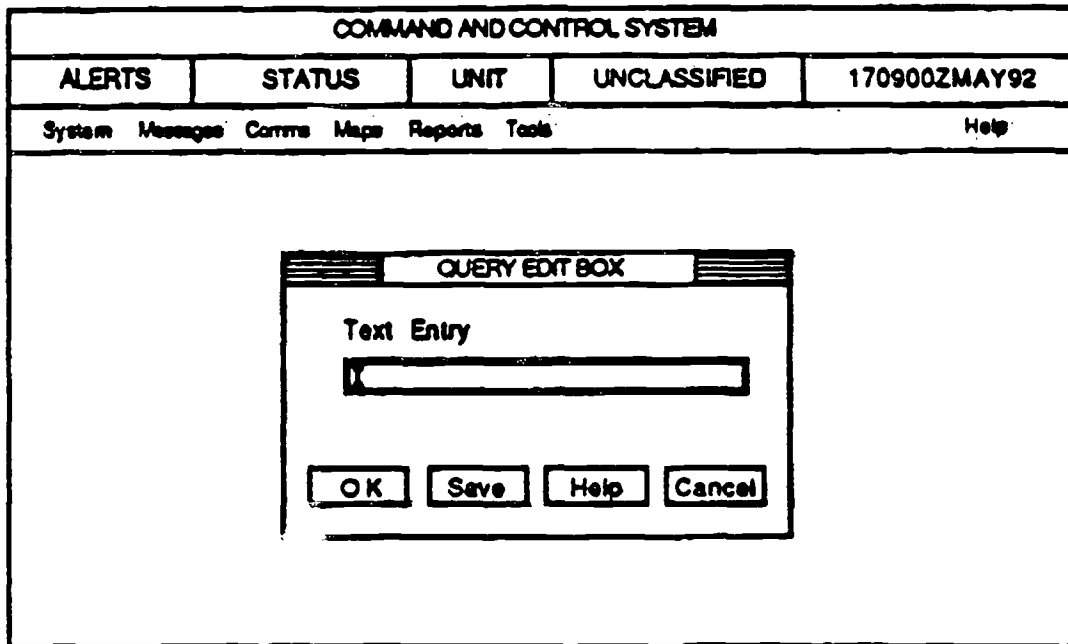


FIGURE 11.3: Sample Text Editing Box

- a. CUT should allow the user to remove the selected text and place it in a clipboard.

- b. **COPY** should allow the user to duplicate selected portions of text and place them in a clipboard.
- c. **PASTE** should allow the user to place text from the clipboard into the current text.
- d. **CLEAR** should remove all characters currently in the text-editing box.
- e. **SEARCH** should allow the user to locate a word or group of characters in the text-editing box.
- f. When used in conjunction with **SEARCH**, **REPLACE** allows the user to replace a word or set of characters with another word or set of characters.
- g. **SPELL CHECK** should check the words in the text-editing box against a dictionary of recognized words. This function also should check textual commands to assure correct spelling and syntax.

11.2.5.4 Query Formulation Commands

Major tasks performed by queries include extracting, manipulating, and performing calculations on tabular data, including creating tabular results and new tables. Query applications should be able to build functions as needed for developing application programs, as well as update and maintain tables.

- a. **SELECT** should provide a means of identifying fields to appear in the query results.
- b. **COMPILE** should generate an executable function and check for correctness.
- c. **RUN** or **DO QUERY** commands cause execution of the query. The application should monitor the execution with prompts for input and error recovery.
- d. The **SHOW** command should allow various presentations of a tabular result and could be used to present a preview of the results of a query or report.
- e. **MODIFY** should allow the user to make changes to the query definition of an already existing query or report. The new query could be saved to a file or as a report, if desired.
- f. **SAVE** should allow repeated use or modification of a query. Store the queries

in a file with a unique extension, such as ".QRY."

11.2.5.5 User-Friendly Searching

- a. Abbreviations should be significantly shorter than the original word or mnemonic. Truncation is the preferred form of abbreviation. The application should allow both the abbreviation and the full term.
- b. The application should automatically complete a search term (opposite of truncation) as soon as it recognizes that the portion of the term entered is unique in the index of search terms. The application should stop the user from typing once the search is uniquely identified.
- c. Because even simple queries can overload the computer, the computer should inform the user of the problem and prompt user input to terminate the query or continue.

11.2.5.6 Features

- a. Provide an interactive program that allows the user to navigate through the database. The BROWSE function is especially helpful when queries would be too lengthy to run interactively.
- b. Provide facilities to format the results of queries as reports.
- c. Ensure that the application provides the ability to view the list of words and phrases available for searching and term variations, including a link to a database thesaurus to suggest search terms.
- d. Use parsing which is the process of deciding how the field will be entered into the search index. Parsing decisions have a direct impact on how a database can be searched and provide flexibility in searching, regardless of how fields were parsed.
- e. Use proximity searching which provides the ability to search words in a positional relationship from word index fields, such as titles or abstracts. The words should be either in a specific order or independent of order. For example, the words "query" and "formation" could be searched in the same field.
- f. Ensure that the application provides the use of Boolean logic, including the use of the logical operators AND, OR, and NOT. It should prompt the user for sets

consisting of search terms and combine (intersect) the sets. The search can then be completed as a combined (union) set. The application also should allow interactive editing of queries.

- g. Ensure that the application provides set building as a means of performing the search in a series of steps, then views the records that answer a query as a set defined by the query.
- h. Include range searching in the application. This type of search should be based on an ordered sequence using FROM and TO.
- i. Ensure that the application allows the user to specify the fields to search, because limiting a search to particular fields may speed the search.
- j. Use a controlled vocabulary of natural language terms. This helps novice users formulate queries.
- k. Ensure that the application facilitates selecting search terms from key words in records. Then, the interface can display these terms and prompt user selection. For example, the application could rank additional search terms by frequency of appearance in a retrieval set and provide them in ranked order.
- l. Provide the capability for the system to search on specific data field values input by the user. The application should provide a list of possible field values from which users select.
- m. Ensure that the application is able to order the field values in a reasonable way, such as alphabetically or from greatest number to the smallest.
- n. Ensure that the application provides a crossfile search, which will obtain the number of references in all potential databases for the search terms or search profile.

11.2.6 MULTIPLE LEVELS

User-friendly features and requirements differ for the novice and experienced user. Because the novice will become a more experienced user, the HCI needs to change to suit the evolving needs of the user and the demands of users who have different levels of expertise.

11.2.6.1 Accommodate Novice and Experienced Users

Multiple levels of interaction are necessary to accommodate the varying levels of experience.

- a. Users should be able to change levels at any time during a session.
- b. A tutorial mode should be available when possible.
- c. Offer context-sensitive HELP on request at all levels.

11.2.6.2 Novice Users

The level of computer knowledge required of a novice user should be minimal. The application should be easy to use, provide familiar terminology, and allow the user to begin work with little training. To be used effectively, an application should not depend on a complex command language. However, this ease of use may require a loss of power and flexibility.

- a. An interface for novices may contain only a subset of the search capabilities. This system may be a scaled-down version of a more comprehensive program. In addition, these interfaces may require fewer searchable fields, so the system may not attain the same specificity or variety of search techniques.
- b. The computer software should prompt the novice user to select from a list of options. The interface should provide an explanation of the options presented.
- c. The interface for novices should have a simplified command structure using fewer and more easily understood commands.
- d. Mnemonic selections are preferred over numbered selections.
- e. The system design should strive for intelligent interfaces between naive users and search systems. Two main components of an intelligent front-end are forms and graphics. Menus and data forms can control the flow of the application, and graphics can be used to provide a visual readout of the data.

11.2.6.3 Experienced User

Experienced users can accommodate comprehensive versions of query applications.

- a. The application designed for the expert user can reduce computer overhead by providing less detailed on-line information.
- b. The application should allow the experienced user to enter multiple commands to speed the dialog.

11.3 EMBEDDED TRAINING

The interface of the optimally designed computer program should be designed and tested such that no user assistance is needed. However, because of differences between humans and computers, the variety of task demands, and the ever-present human tendency to make errors, assistance is needed. People differ in computer experience, patience level, learning style, reasoning ability and style, and numerous other characteristics. At the same time, sophisticated computer systems and software programs are often highly complex but still retain the requirement to be highly usable without requiring extensive training or technical expertise. Assistance programs offer one of the primary methods used by designers to achieve a high degree of usability.

User assistance is commonly offered through on-line help, documentation, and on-line training. The distinction between on-line help and on-line training is often blurred. For the purposes of this *Style Guide*, on-line help refers to assistance for a specific problem, function, command, or term. On-line training programs focus on process; they offer instruction.

On-line training programs may exist completely embedded within the application software, separately as an application, or as a combination of both. The on-line training program also may be executed by some form of supplemental component (e.g., strap-on [video disk player] or plug-in [floppy disk]). Though many guidelines apply to both embedded and supplemental training, the interface guidelines presented in this section pertain specifically to embedded training.

The guidelines also apply to a range of embedded training formats and capabilities including:

- Fixed format provides the same information regardless of what the user has done.
- Context-sensitive format depends on what users are currently trying to do or on the context in which they are working.
- Prompting intervenes or prompts automatically if a user proceeds incorrectly.
- Dialog allows users to obtain assistance through natural language interaction.
- Adaptability keeps track of a user's operation and provides appropriate help or training based on the user's operation, for example, intelligent tutoring systems (Kearsley 1986).

The embedded training guidelines included in this section are derived from the results of empirical research, reported computer training experience, and experts' recommendations. Guidance for embedded training interface design appears under a variety of types of on-line training: Computer-Assisted Instruction (CAI) and Intelligent Computer-Assisted Instruction (ICAI), Computer-Based Training (CBT), Intelligent Tutoring Systems (ITS), Embedded Training (ET), coaching; Electronic Performance Support Systems (EPSS), and guided discovery, among others.

On-line training strives to support learning how to use an application. However, it conflicts with the user's primary task, because consulting a training program interrupts work in progress. This conflict may cause new users to skip training altogether or to select immediate task help without furthering their overall understanding of the system (Grice 1989; Hackos 1991; Horton 1990). Two crucial factors in determining whether or not users accept and use embedded training are a well designed, intuitive interface and the opportunity to practice. Each of the embedded training guidelines addressed in this section assumes a basic set of objectives for assisting users: consistency, efficient use of capabilities, minimal memory load on users, minimal learning time, and flexible support of different users.

The goal of embedded training interface design is to ensure users can obtain answers to their questions with maximum efficiency,

maximum accuracy, and minimum additional memory requirements. An embedded training program should answer the following types of questions:

- **Goal-oriented:** What types of things can I do with this program?
- **Descriptive:** What is this? What does this do?
- **Procedural:** How do I do this?
- **Interpretive:** Why did that happen? What does this mean?
- **Navigational:** Where am I?
- **Choice:** What can I do now?
- **History:** What have I done? (Baecker and Small 1990; Gery 1991; Laurel 1990).

The manner in which assistance is provided affects the ability of users to learn and transfer that learning to other situations. Research in instruction and on-line documentation has identified basic concepts and practices that support learning and transfer. Central among these concepts relating directly to embedded training are:

- opportunity to practice
- readability
- user control - perceived and actual
- learning mode - visual (graphics and text)
- advance organizers.

Specific guidelines related to these concepts appear in embedded training components and instructional presentation guidelines (see Paragraphs 11.3.4 and 11.3.6).

Much of the relevant research and development work from which these guidelines were developed comes from individual demonstration and limited distribution systems. Significantly less empirical research has explored the behavioral issues pertaining to on-line training or advice-giving systems. On-

line training experts call for additional behavioral research, for example, effects of feedback timing, preferences for and effectiveness of different on-line training components, suitability of media presentation (animation, text, sound), and the effects of system-initiated intervention. The current research focus has moved from building systems capable of detecting all possible errors and misconceptions to building an empathetic partner that chooses among several forms of interaction based on the content of the task and needs of the user.

11.3.1 GENERAL

A strong embedded training interface provides users with an understanding of the training program and the linkage between the application program and the training program. The guidelines in this section address user orientation and the linkage between application and embedded training programs.

11.3.1.1 Initial Use Overview

Provide first-time users of embedded training an overview of the embedded

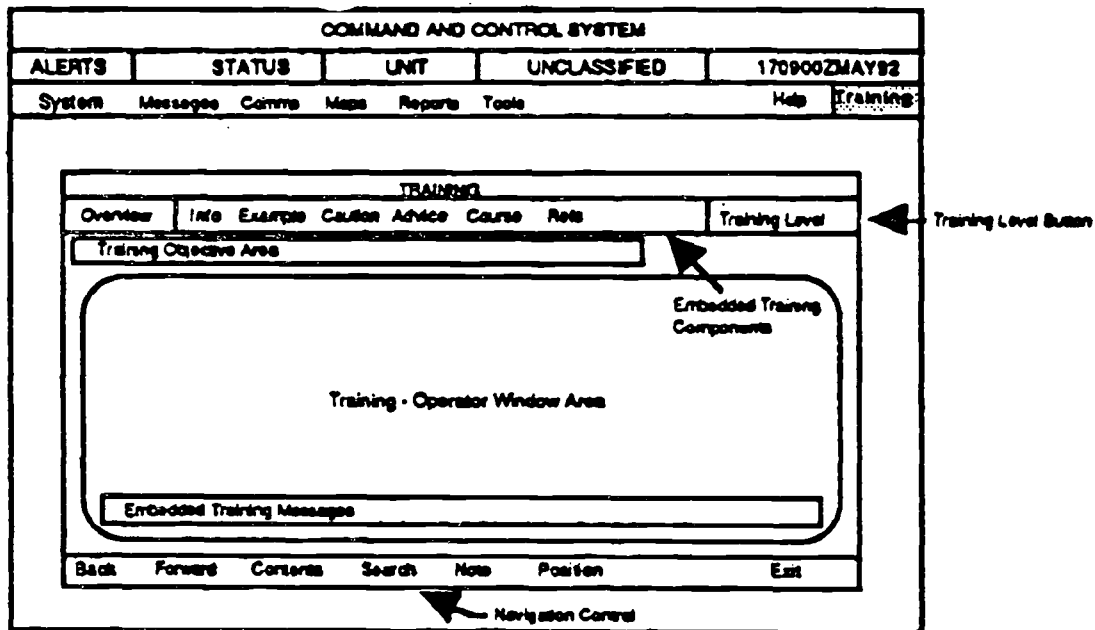


FIGURE 11.4: Simplified Prototype Embedded Training Screen

training program. This orientation should convey what the embedded training

achieves by combining text and graphics (animated or static).

11.3.1.2 Positive User Attitude

Build positive user attitudes and increase use of the embedded training by ensuring that the interface maintains a positive tone and does not evaluate the user's performance when practicing and experimenting. Ensure the system messages do not blame the user, and avoid implying that the computer is

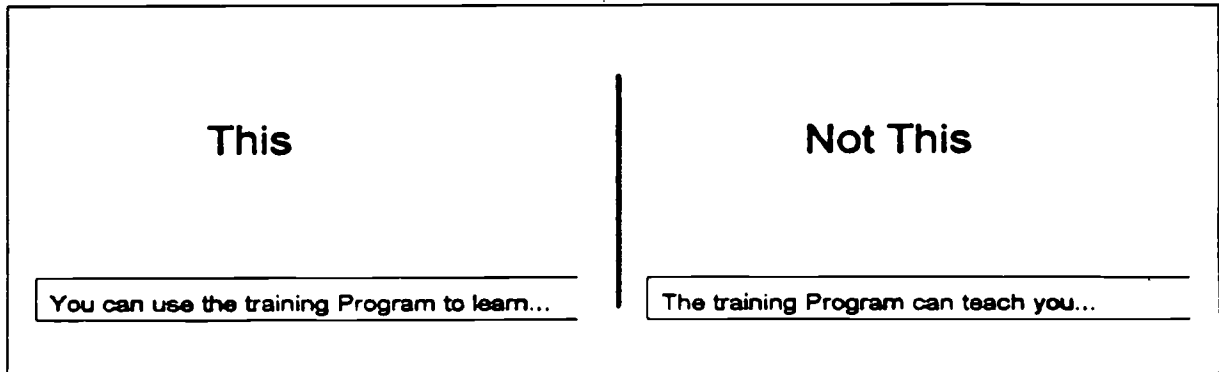


FIGURE 11.5: Examples of Implying Human Characteristics

human. See Figure 11 5.

- a. Do not use personalized messages, as they interrupt and often annoy users.
- b. The effectiveness of the embedded training is related directly to the accuracy of the embedded training information.
- c. Avoid personalization (i.e., "You did a good job, Sam") and personal recognitions (including even simple statements, such as "Excellent!").

11.3.1.3 Availability of Embedded Training

Provide embedded training programs to users at all points during use of the application, except where it would interfere with time-critical operations.

11.3.1.4 Accuracy of Embedded Training

Ensure embedded training is accurate, reflects the most current form of the application, and is updated in response to changes in the application. When changes occur in application procedures or critical operations, ensure users are notified. In addition, consider providing users with an option to see new and

revised information (i.e., by selecting "News").

11.3.1.5 Moment of User Need

Provide training support at the moment the user needs it whenever possible.

11.3.1.6 Embedded Training Browsing

Allow users to work with the embedded training independent of the application, to accommodate user browsing.

11.3.1.7 Return to the Application From Embedded Training

Ensure users can return to the application from any point within the embedded training with a single action (e.g., keystroke, command, point and click) without shutting down either system.

11.3.1.8 Restore Application Screen

When users exit the training program, restore the application screen to the state that existed prior to the request.

11.3.1.9 Restore Embedded Training

When training is interrupted (e.g., system failure, tactical operation requirements) or users exit before completion, offer them the opportunity to return to the position in the embedded training that existed before the interruption.

11.3.1.10 Protection from Hazardous or Destructive Actions

Prohibit users from accidentally activating hazardous events (e.g., mine field activation) and destructive control actions (e.g., accidental erasure or memory dump) during the embedded training.

11.3.1.11 Application Screen Protection

Ensure embedded training commands do not alter or destroy application screen data.

11.3 1.12 NonInterference During Critical Operation

Prohibit system-initiated embedded training interruption of the primary application during a critical operation.

11.3.1.13 Notification of Critical Operation

Ensure the user is notified of incoming critical application information (e.g., tactical operation input).

11.3.1.14 Multiple Stations

If the application system has multiple stations, ensure that stations using the embedded training have no effect on the stations performing an operational task.

11.3.1.15 Context Sensitivity

Make the training context-sensitive; that is, wherever possible, the training should depend on where the user is in the application or on the general nature of the content of the application.

11.3.1.16 Consistent Application Interface

Provide the greatest possible consistency between the application interface and the embedded training interface to ensure a smooth transition between platforms and to minimize the user's learning requirements (e.g., terminology, displays, commands).

11.3.1.17 Inconsistent Interface Assistance

Provide assistance if the embedded training interface is substantially different from the application systems operations or when the embedded training interface has complex features that might need to be explained.

11.3.2 ADAPTATION TO USERS

Users vary in many ways, including computer experience, domain experience (program content - e.g., command and control), learning style, preferred work style, and immediate task demands. Adapting the embedded training interface to the user's characteristics and preferences will encourage use of embedded training and, consequently, should increase user efficiency with the application. See Figure 11.6.

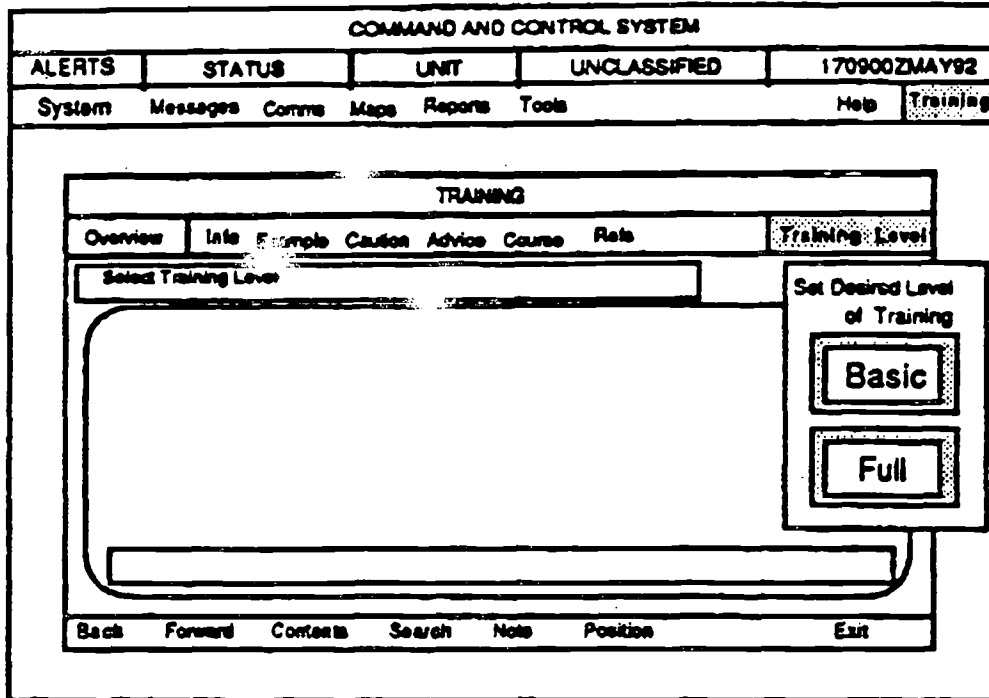


FIGURE 11.6: User Selection of Training Level

11.3.2.1 User Control Over Level of Difficulty

Accommodate the differences in computer experience by allowing users to select the level and type of assistance. The ability to select is important because users may be novices in some areas, casual operators in others, and experts in still others. Reducing the complexity of the training interface for beginners simplifies the demands of learning.

- a. Allow novice users and/or first-time users to select a restricted capability interface that blocks features and allows only basic feature operation (e.g., in word processing - creating, editing, and printing).
- b. Provide novices only the necessary information, but allow them access to all of the capabilities by direct request.
- c. Offer experts assistance in using the system more efficiently (e.g., shortcuts, limitations, complex operations).

11.3.2.2 Learning Structure

Allow the user to select a type of learning structure. This accommodates the user's individual needs for information and practice. Learning structure types may be:

- a. discovery - undirected exploration or browsing
- b. guided/supported discovery - directed exploration
- c. structured - menu identifies options explicitly and provides implicit cues.

11.3.3 EMBEDDED TRAINING COMPONENTS

An embedded training system can integrate several resource components to support users while they perform their jobs. Embedded training programs may provide an information database (infobase), common errors, examples and scenarios, interactive advice, internal cross-referencing, expert system-initiated training, and formal courseware.

11.3.3.1 Multiple Components

In addition to the immediate context-sensitive assistance, offer users multi-component training that is easy to specify and access (e.g., scenarios, examples, information databases, off-line references, common problems, and/or coaching).

11.3.3.2 Information Database Component

Provide users an interactive information database containing both conceptual and task-oriented information.

11.3.3.3 Reference Component

Provide a reference component that includes all on-line resources, as well as system- and job-related, off-line resources.

11.3.3.4 Examples Component

Offer users the opportunity to practice using common examples in an exploratory or guided mode, which would allow users to work through the steps required to perform a specific task.

- a. Encourage user experimentation (i.e., "What would happen if...") by making it easy for them to exit an application, practice, then return to the unaltered application position.
- b. If users explore a problem within the application, protect the application system with an UNDO command requirement.
- c. Clearly distinguish between the exercise and the application to minimize possible confusion arising from switching back and forth between operation modes (e.g., highlight or shadow the practice session).
- d. Avoid demonstrations and exercise summaries if they do not provide opportunities to practice the procedure or function.

11.3.3.5 Advisor or Coaching Component

Provide an embedded training component that advises or coaches users in solving problems. This may make users aware of enhanced system operation and may also be used in response to user request, system recognition of suboptimal user performance, or complex tasks. A system can coach users through tasks by presenting a series of questions and recommending a course of action based on the responses (Figures 11.7 a-c).

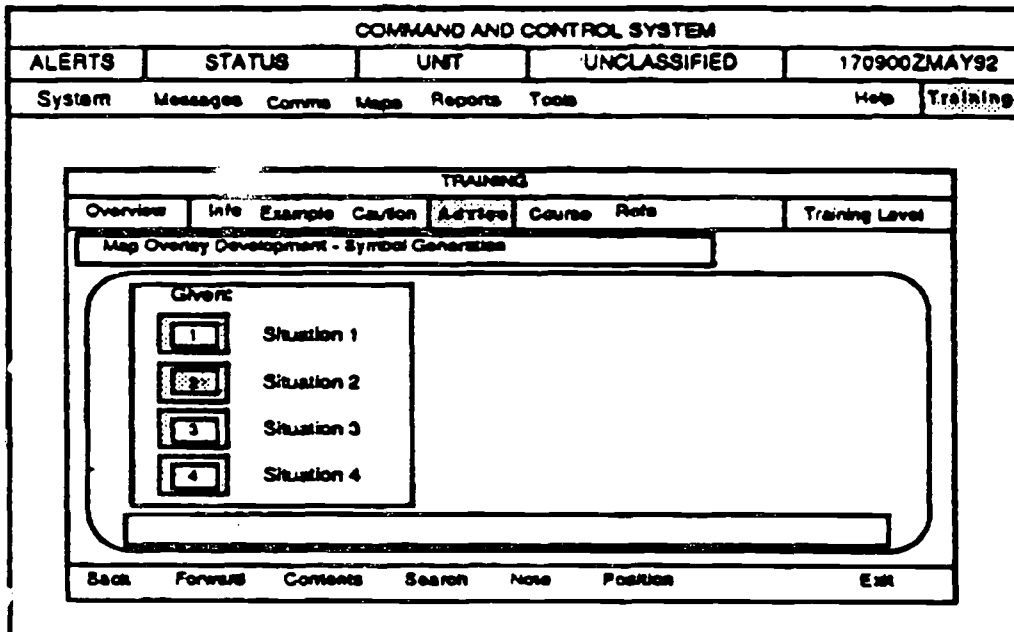


FIGURE 11.7a

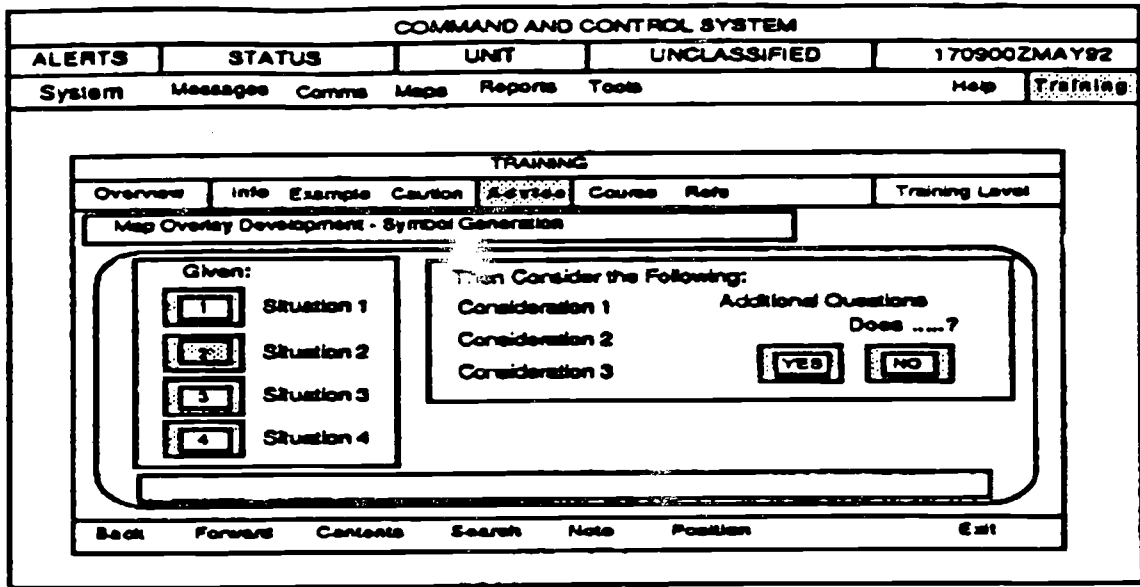


FIGURE 11.7b

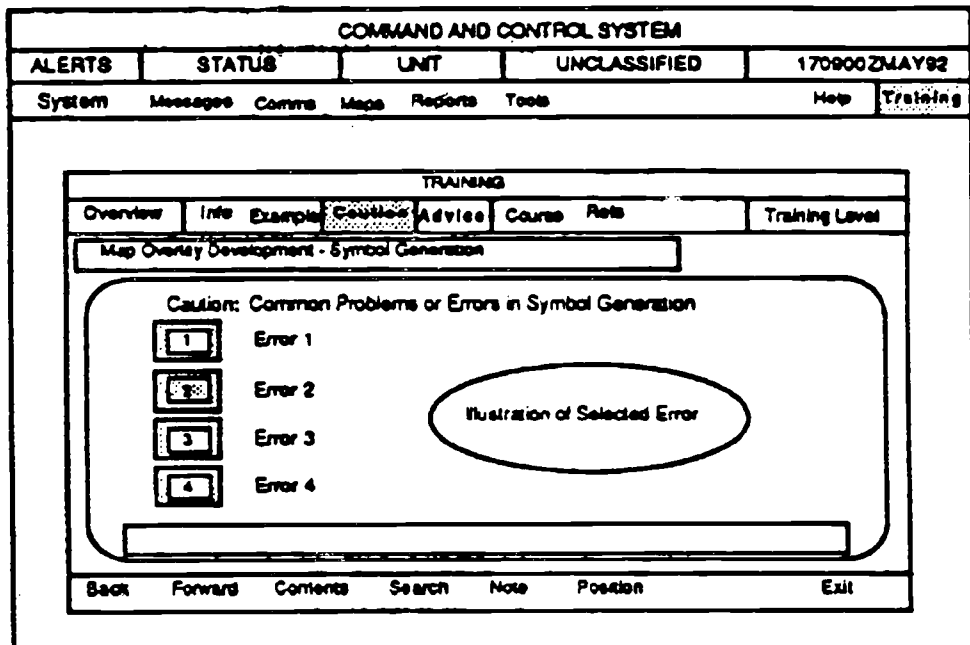


FIGURE 11.7c

FIGURE 11.7: Example of How an Advisor Can Guide Users
11-47

11.3.3.6 Common Errors Component

Provide users with a context-similar, embedded training component that shows common user errors or "Cautions" associated with a given approach or task procedure. See Figure 11.8.

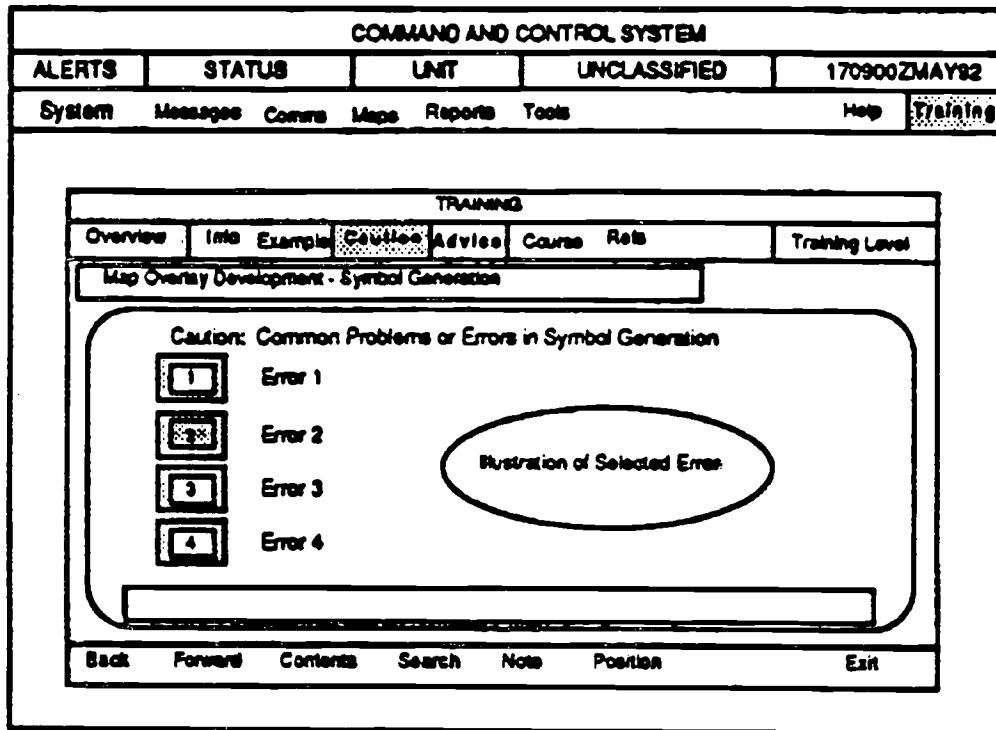


FIGURE 11.8: Example of "Cautions" that Identify Common Errors

11.3.3.7 Record Keeping

Records of user interaction with the embedded training can be helpful to users, supervisors, and system designers. Using embedded training records requires careful planning to avoid threatening users. Users are more likely to experiment and practice if they feel their errors will not be seen by others.

- a. Allow the user to record the path through a process and/or the training modules completed successfully or unsuccessfully. This will aid in later reference or experimentation.

- b. If records of user training sessions are stored, ensure the privacy of users is protected by storing their records as anonymous files.
- c. If the exercise or courseware module will be used for evaluation purposes, give users prior notification. Explicitly state the criteria for evaluation.

11.3.4 INSTRUCTIONAL STRUCTURE

Both the size of instructional unit (granularity) and control of instructional sequence affect the efficiency and attitude of the user.

11.3.4.1 Granularity

Structure the embedded training components into "single learning episodes," small enough and homogeneous enough to be learned as single units. This enables users to select the particular section or subtopic within a component for which they desire assistance.

11.3.4.2 System-Controlled Sequences

For novice users and for embedded training that deals with critical or hazardous procedures, the system should direct user movement through the procedures.

11.3.4.3 Sequence Control for Experienced Users

Provide experienced users with the flexibility to move through the steps of a procedure sequentially or to move directly to any specific step or resource point.

11.3.5 INSTRUCTIONAL PRESENTATION

The manner in which assistance is provided affects the ability of users to learn from the instructional experience and to transfer that learning to other situations. The following statements outline interface guidelines for instructional presentation.

11.3.5.1 Combined Media Presentation

Present the embedded training using a combination of media, graphics, and natural language, where appropriate. Graphic media aid in visualizing

significant patterns, whereas natural language text conveys the meaning and significance of the visualization. See Figure 11.9.

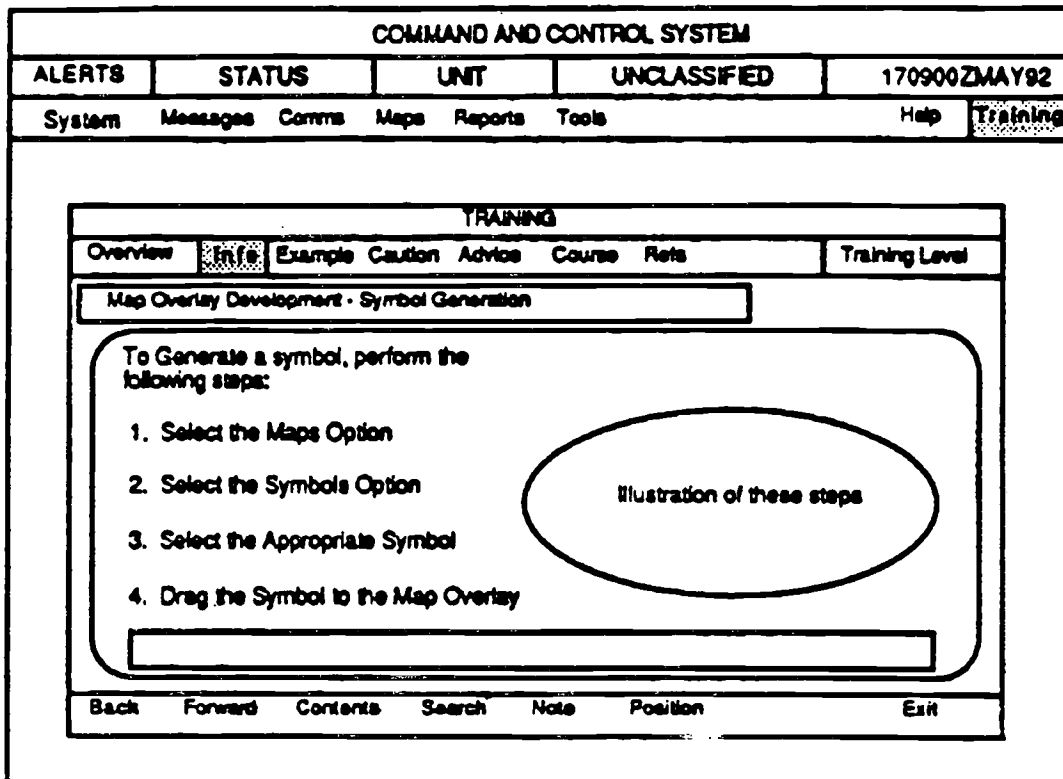


FIGURE 11.9: Combined Graphic and Natural Language Presentation

11.3.5.2 Graphics for Method-Based Knowledge

Offer users flowchart diagrams that provide an overview conveying method-based knowledge, consisting of a series of procedural steps and decisions.

11.3.5.3 Reading Requirements

Keep reading requirements to a minimum. Users prefer to read text in print and often will not read text on a screen that exceeds even a few sentences in length.

11.3.5.4 Advance Organization

If the component will be used for knowledge training, provide cues and overviews that orient users unfamiliar with the embedded training content and/or process through brief descriptions of scenarios and exercises or courseware outlines. Stated objectives are an important feature for novice users.

- a. Provide users a brief statement of the exercise objective. The statement should refer to the primary purpose of the embedded training request.
- b. Clearly identify each embedded training module, stating objective, content, and, where appropriate, the number of subsections and estimated completion time.
- c. Remind users of the purpose of their request for assistance. See Figure 11.10.

11.3.5.5 Printing

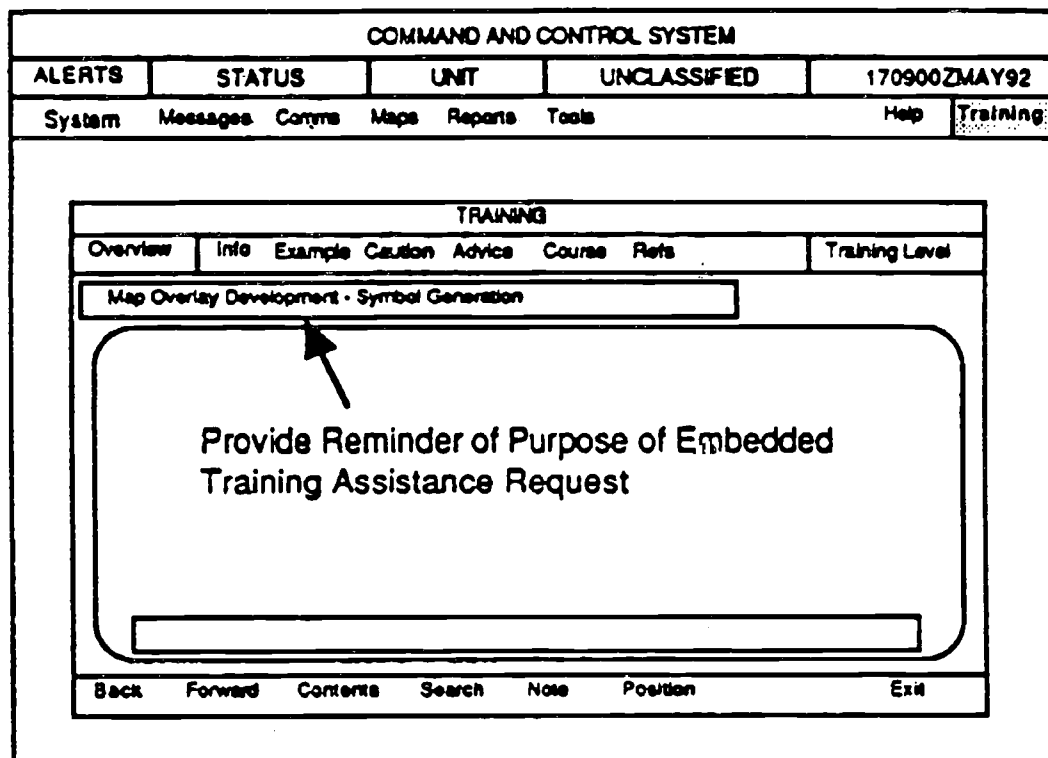


FIGURE 11.10: Example of Assistance Request Reminder

Provide users the ability to print embedded training content - ranging from screen displays to courseware, information to study further, or for future reference, and/or to print the displayed training material.

11.3.5.6 Fidelity

Adjust the level of training content and presentation fidelity to match the training:

- a. low fidelity for initial training, simple data, and easy process
- b. high fidelity for unusual processes, hazardous events, or difficult processes.

11.3.5.7 Simplicity

Give users simple answers to simple questions. If the answer is long or complex, offer a summary and options to request additional guidance.

11.3.5.8 Verification

Allow users to verify or confirm selected options, solutions, and commands. This allows them to evaluate the completeness of a process or task and the accuracy of their approach without having to sort through extraneous material.

11.3.6 ACCESSING TRAINING

11.3.6.1 Displayed Embedded Training Availability

Display the command, icon, or function key used to access training throughout the application to remind the user of training availability.

11.3.6.2 Access Via Training Icons

Allow users to access the embedded training directly by selecting an embedded training icon and moving to the point of user need. For example, a user could activate embedded training and select a "Cautions" component icon, move to the point where assistance is needed, click, and receive additional information without exiting the application. See Figure 11.11.

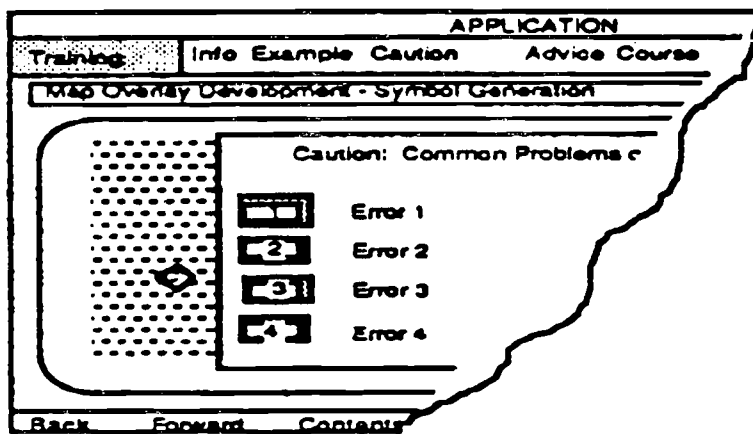
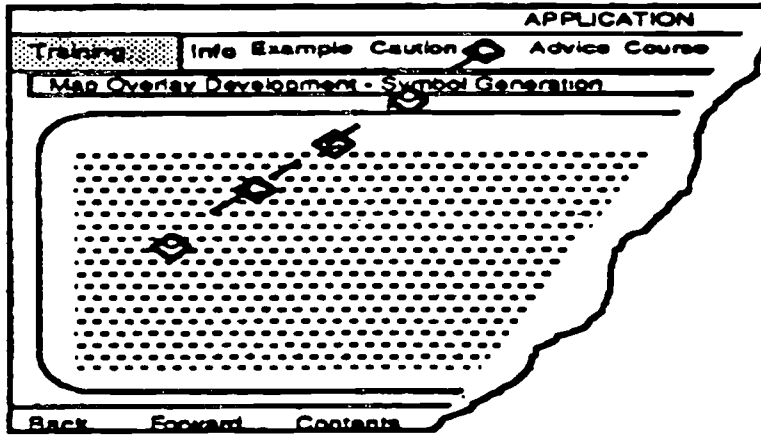
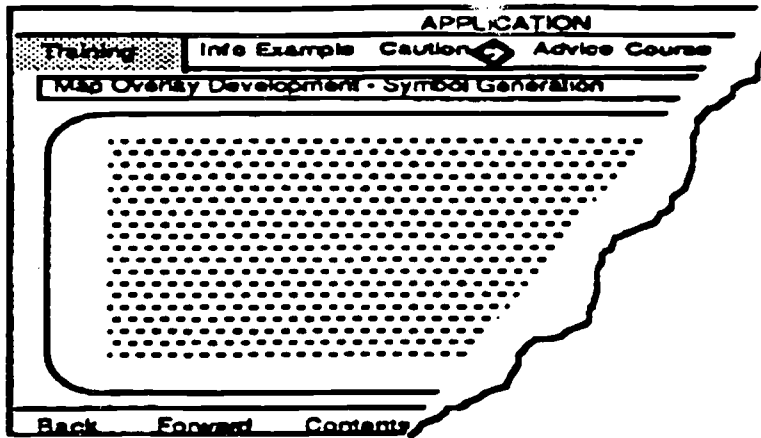


Figure 11.11: Using an Icon to Access Embedded Training Directly

11.3.6.3 Structured Menu

When using a structured menu to access the embedded training, allow users to add to or change existing embedded training messages (e.g., add terms to the menu using an "ADD" function). If users are allowed to customize menus, the original menu must be protected (e.g., log-on files for individual users).

11.3.7 SCREEN DISPLAY

11.3.7.1 Complete Display

The content of each screen should stand on its own; do not require users to refer to a previous screen within a module to recall essential information. For example, if users need to enter identical information on a series of screens, the system should automatically enter the appropriate information, repeat the information on each screen, or prompt users to record the information.

11.3.7.2 Graphics

Select uncomplicated graphics that portray the functional objective clearly, omitting nonessential visual detail.

11.3.7.3 Window Placement

Display the assistance in windows that do not completely obscure the application's "critical" navigation buttons, operational icons, or the status message line or window.

11.3.7.4 User Window Control

Allow users to resize and reposition overlapping windows. This will allow users to see the portions of the application with which they are most concerned.

11.3.8 TECHNICAL COMMUNICATION/WRITING STYLE

Phrase embedded training topics, messages, and menu options in the active voice. Phrase task-related terms to refer to the learning task (e.g., "Creating and modifying fields" instead of "Fields").

11.3.9 MOBILITY/NAVIGATION

11.3.9.1 Mobility Within the Embedded Training

Allow users to move among embedded training components freely:

- a. without returning to the top of a central hierarchy
- b. without exiting the current embedded training component
- c. without having to proceed through a preset path
- d. without having to step through introductory material

11.3.9.2 Embedded Training Navigation Button Display

Display embedded training navigation "buttons" in each embedded training screen for controlling movement between and within modules. See Figure 11.4.

11.3.10 ERROR FEEDBACK

11.3.10.1 Immediate Feedback

Provide feedback to users in a timely manner, adjusted to the users' expertise.

- a. When practice requires multiple steps, provide users immediate feedback to avoid a series of incorrect actions.
- b. For novices and for uncomplicated problems, offer immediate feedback that includes a suggested next best step.

11.3.10.2 Context-Similarity Feedback

Provide feedback to users in a form similar to the application, product, or outcome (e.g., an error in equipment setup may be illustrated by correctly configured equipment rather than by a checklist, menu, or even natural language message).

Allow users to move among embedded training components freely:

- a. without returning to the top of a central hierarchy
- b. without exiting the current embedded training component
- c. without having to proceed through a preset path
- d. without having to step through introductory material

11.3.9.2 Embedded Training Navigation Button Display

Display embedded training navigation "buttons" in each embedded training screen for controlling movement between and within modules. See Figure 11.4.

11.3.10 ERROR FEEDBACK

11.3.10.1 Immediate Feedback

Provide feedback to users in a timely manner, adjusted to the users' expertise.

- a. When practice requires multiple steps, provide users immediate feedback to avoid a series of incorrect actions.
- b. For novices and for uncomplicated problems, offer immediate feedback that includes a suggested next best step.

11.3.10.2 Context-Similarity Feedback

Provide feedback to users in a form similar to the application, product, or outcome (e.g., an error in equipment setup may be illustrated by correctly configured equipment rather than by a checklist, menu, or even natural language message).

11.3.10.3 Error Identification

Provide specific feedback that identifies errors rather than assigns a score.

11.3.10.4 Tone of Error Message

Provide error messages that are constructive and neutral in tone; avoid messages that suggest a judgement of the user's behavior. See the examples in Figure 11.12.

This	Not This
NEUTRAL:	JUDGEMENTAL:
<div style="border: 1px solid black; padding: 2px; width: fit-content; margin: auto;">The System cannot....</div>	<div style="border: 1px solid black; padding: 2px; width: fit-content; margin: auto;">Invalid Number: Entry must be 4 digits</div>

FIGURE 11.12: Examples of Tone in Error Message

11.3.10.5 System-Initiated Error Feedback

- a. When a response to a system-initiated query or recommended action is required, the system should provide users the opportunity to stop and think (i.e., consider other options, recall past experiences, weigh problem solutions) without premature system interruption. For example, the system should avoid over-prompting and unwanted problem resolution. The system could offer users the ability to place the system initiation on hold or cancel an upcoming intervention.
- b. Provide novice users with prompts identifying probable next-step errors.
- c. Use control blocking sparingly (e.g., to protect a system from accidental hazardous actuation and system destruction).
- d. Allow novice users to select an error-blocking option that limits errors.
- e. Give users the option to block temporarily the system-initiated error feedback or instruction.

- f. If the system automatically corrects some errors (e.g., replacing an out-of-bounds parameter), ensure users are notified of the corrections (e.g., by a message and highlighting of the corrected information). See Figure 11.13.

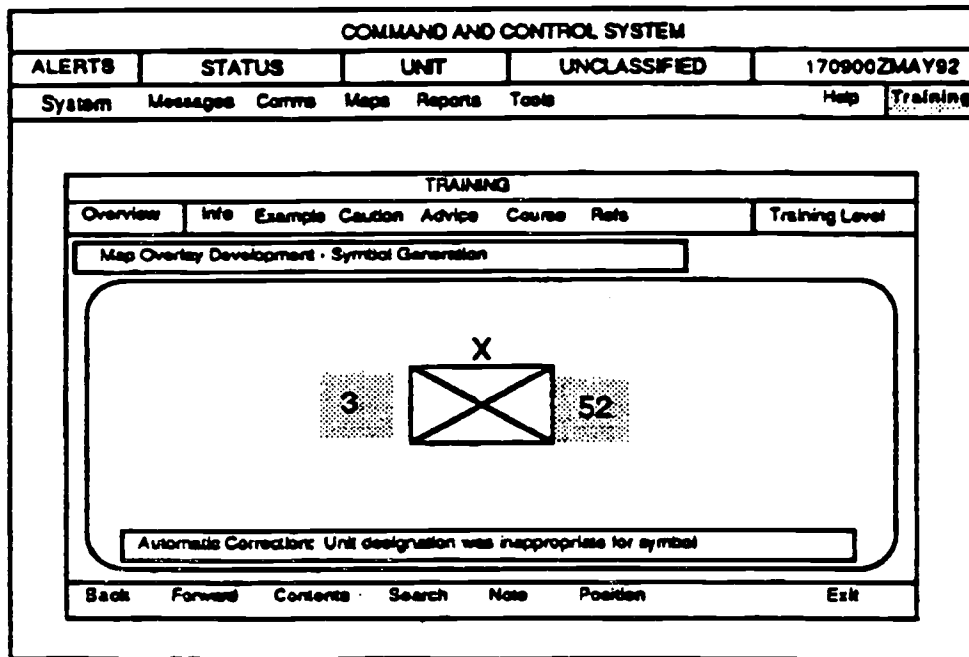


FIGURE 11.13: Example of Automatic Correction Notification and Identification

- g. Allow experienced users to select automatic system error correction without requiring their confirmation. This option assumes that mundane errors made by experts are a result of minor actions, such as misstriking a key, command, or icon.
- h. Blocking access to system functionality can be very frustrating and can be a result of a misdiagnosed error or correct, but uncommon, approach.
- i. Avoid user confusion that may result from automatic system error correction.

11.3.11 ABILITY TO MODIFY

11.3.11.1 Additions to the Embedded Training

Provide users the capability to add to, but not modify, the original training system. If the embedded training allows individual user modifications, the

original application and embedded training should be protected (e.g., separate log-on files for each user).

11.3.11.2 Multi-User Systems

On multi-user systems, permit the individual user to store and reference additions in a individual file.

11.3.11.3 Annotation

Permit users to annotate a copy of the training program (i.e., examples, pitfalls, process notes, references, etc.).

11.3.11.4 Annotation Search

Provide users the ability to search an annotation log.

11.3.11.5 Icons Used to Designate Annotations

Use icons to designate the position of an annotation in the embedded training program (e.g., a user's example, caution, or additional reference). See Figure 11.14.

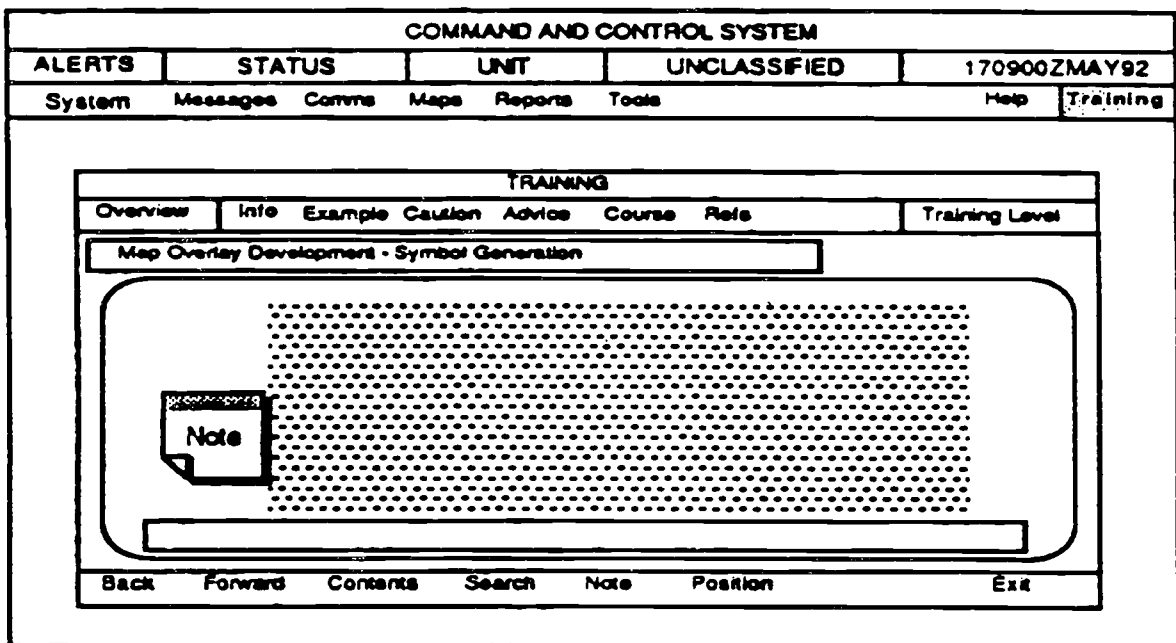


FIGURE 11.14: Example of Identification Annotation Position

REFERENCES

<u>Paragraph</u>	<u>Reference</u>
11.1.1.1a	Zachary (1988) pp. 997-1030; McKeown et al. (1991); Ehrhart (1990); Andriole and Adelman (1990)
11.1.1.1b	Andriole and Adelman (1990); Zachary (1988) pp. 997-1030; Walrath (1989)
11.1.1.1c	Zachary (1988) pp. 997-1030; Walrath (1989); Andriole and Adelman (1990); McKeown et al. (1991); Ehrhart (1990)
11.1.1.1d	McKeown et al. (1991)
11.1.1.2a	Zachary (1988) pp. 997-1030; Lysaght et al. (1988)
11.1.1.2b	McKeown et al. (1991); Zachary (1988) pp. 997-1030; Lysaght et al. (1988)
11.1.1.2c	McKeown et al. (1991); Lysaght et al. (1988)
11.1.1.3	McKeown et al. (1991); Lysaght et al. (1988)
11.1.1.4a	Holtzman (1989); Lysaght et al. (1988); McKeown et al. (1991); Walrath (1989)
11.1.1.4b	Walrath (1989); McKeown et al. (1991); Holtzman (1989)
11.1.1.4c	McKeown et al. (1991); Walrath (1989); Lysaght et al. (1988); Klien and MacGregor (1988)
11.1.1.4d	Holtzman (1989); Klien and MacGregor (1988); McKeown et al. (1991); Walrath (1989)
11.1.1.5a	Holtzman (1989)
11.1.1.5b	Holtzman (1989)
11.1.1.5c	McKeown et al. (1991)
11.1.1.5d	McKeown et al. (1991)
11.1.1.6a	Walrath (1989)
11.1.1.6b	Walrath (1989)
11.1.1.6c	Walrath (1989)
11.1.1.6d	Walrath (1989); Gordon (1988) pp. 55-59
11.1.2.1a	Gordon (1988) pp. 55-59; Thierauf (1988); Main and Paulson (1988)
11.1.2.1b	Thierauf (1988)
11.1.2.1c	Main and Paulson (1988)
11.1.2.1d	Gordon (1988) pp. 55-59
11.1.2.2	Thierauf (1988)
11.1.2.2a	Thierauf (1988); Holtzman (1989); Schwartz (1983) pp. 13-17; Andriole and Adelman (1990); LeMay (1988) pp. 227-229; Finke and Lloyd (1988) pp. 170-193

REFERENCES (Cont.)

<u>Paragraph</u>	<u>Reference</u>
11.1.2.2b	Urban (1990); Andriole and Adelman (1990); Thierauf (1988); McCann (1988); Main and Paulson (1988)
11.1.2.3a	Urban (1990); Andriole and Adelman (1990); Thierauf (1988); Walrath (1989)
11.1.2.3b	Thierauf (1988); Tannenbaum (1990) pp. 54-59; Bidgoli (1989) pp. 27-34; Andriole and Adelman (1990); Urban (1990); Zachary (1988) pp. 997-1030
11.1.2.3c	O'Keefe (1989) pp. 217-226; LeMay (1988) pp. 227-229
11.1.2.4a	Minasi (1990) pp. 13-15; O'Keefe (1989) pp. 217-226
11.1.2.4b	Holtzman (1989); Schmitz et al. (1990) pp. 29-38; Pew (1988) pp. 931-940; Holtzman (1989)
11.1.2.4c	Holtzman (1989); Bidgoli (1989) pp. 27-34; Thierauf (1988); Pew (1988) pp. 931-940; Finke and Lloyd (1988) pp. 170-193
11.1.2.4d	Andriole and Adelman (1990)
11.1.2.4e	Finke and Lloyd (1988) pp. 170-193
11.1.2.4f	Finke and Lloyd (1988) pp. 170-193; Minasi (1990) pp. 13-15; Tannenbaum (1990) pp. 54-59
11.1.3.1a	Lysaght et al. (1988); McKeown et al. (1991); Schmitz et al. (1990) pp. 29-38; Pew (1988) pp. 931-940; Ma et al. (1989) pp. 996-1012
11.1.3.1b	Holtzman (1989); McKeown et al. (1991); Gordon (1988) pp. 55-59; Riedel (1988); McCann (1988); Zachary (1988) pp. 997-1030; Thierauf (1988); Walrath (1989); Ehrhart (1990); Pew (1988) pp. 931-940
11.1.3.1c	McKeown et al. (1991); Osborn and Zickefoose (1990) pp. 28-35; Lysaght et al. (1988)
11.1.3.1d	Holtzman (1989); Riedel (1988)
11.1.3.1f	Holtzman (1989); Ma et al. (1989) pp. 996-1012; Pew (1988) pp. 931-940
11.1.3.1g	Thierauf (1988); Schmitz et al. (1990) pp. 29-38; O'Keefe (1989) pp. 217-226; LeMay (1988) pp. 227-229; McCann (1988)
11.1.3.1h	Gordon (1988) pp. 55-59; Riedel (1988)
11.1.3.1i	Gordon (1988) pp. 55-59; Pew (1988) pp. 931-940; LeMay (1988) pp. 227-229
11.1.3.1j	Pew (1988) pp. 931-940

REFERENCES (Cont.)

<u>Paragraph</u>	<u>Reference</u>
11.1.3.1k	O'Keefe (1989) pp. 217-226; Ma et al. (1989) pp. 996-1012
11.1.3.1l	Schmitz et al. (1990) pp. 29-38; Finke and Lloyd (1988) pp. 170-193; Thierauf (1988); O'Keefe (1989) pp. 217-226; Lysaght et al. (1988)
11.1.3.1m	Walrath (1989)
11.1.3.2.a	Gordon (1988) pp. 55-59; Riedel (1988); McCann (1988); Bidgoli (1989) pp. 27-34
11.1.3.2.b	Gordon (1988) pp. 55-59; McCann (1988); Urban (1990); Zachary (1988) pp. 997-1030; Minasi (1990) pp. 13-15
11.1.3.2.c	Pew (1988) pp. 931-940; Finke and Lloyd (1988) pp. 170-193; Urban (1990)
11.1.3.2.d	Riedel (1988); Holtzman (1989); Pew (1988) pp. 931-940; Finke and Lloyd (1988) pp. 170-193; Main and Paulson (1988); McCann (1988); Thierauf (1988); Urban (1990)
11.1.3.2.e	Finke and Lloyd (1988) pp. 170-193; Urban (1990)
11.1.3.3.a	Lysaght et al. (1988)
11.1.3.3.b	McCann (1988); Ehrhart (1990); Holtzman (1989)
11.1.3.3.d	Lysaght et al. (1988)
11.1.3.4.a	Pew (1988) pp. 931-940; Ehrhart (1990); Schwartz (1983) pp. 13-17; Riedel (1988)
11.1.3.4.b	Holtzman (1989); Gordon (1988) pp. 55-59; Pew (1988) pp. 931-940
11.1.3.4.c	Holtzman (1989); Ehrhart (1990); Pew (1988) pp. 931-940; Tannenbaum (1990) pp. 54-59
11.1.3.4.d	Holtzman (1989); Ehrhart (1990); Finke and Lloyd (1988) pp. 170-193; Riedel (1988)
11.1.3.5.a	Riedel (1988); Crolotte et al. (1980) pp. 1216-1220
11.1.3.5.b	Riedel (1988)
11.1.3.6.a	Thierauf (1988); Pew (1988) pp. 931-940; Ehrhart (1990)
11.1.3.6.b	Hoadley (1990) pp. 120-127; Jarvenpaa and Dickson (1988) pp. 764-775; Urban (1990)

REFERENCES (Cont.)

<u>Paragraph</u>	<u>Reference</u>
11.1.3.6.c	Jarvenpaa and Dickson (1988) pp. 764-775; Thierauf (1988); Ehrhart (1990); Finke and Lloyd (1988) pp. 170-193; Bidgoli (1989) pp. 27-34; Zachary (1988) pp. 997-1030
11.1.3.6.d	Andriole and Adelman (1990)
11.1.3.6.e	Barnett (1990) pp. 1552-1556; Jarvenpaa and Dickson (1988) pp. 764-775
11.1.4.1	McCann (1988)
11.1.4.1.a	McCann (1988)
11.1.4.1.b	McCann (1988)12.4.1.3 McCann (1988)
11.1.4.2	Weingaertner and Levis (1988) pp.195-201
11.1.4.3	Weingaertner and Levis (1988) pp. 195-201; Riedel (1988)
11.1.5.1	Thierauf (1988); Schmitz et al. (1990) pp. 29-38; Gordon (1988) pp. 55-59; McCann (1988)
11.1.5.1.a	McCann (1988)
11.1.5.1.b	Holtzman (1989)
11.1.5.2	Bidgoli (1989) pp. 27-34
11.1.5.3	Urban (1990); Gordon (1988) pp. 55-59
11.1.6	Mittal (1985) pp. 32-36
11.1.6.1	Pew (1988) pp. 931-940
11.1.6.1.a	Osborn and Zickefoose (1990) pp. 28-35; Mittal (1985) pp. 32-36; Schwartz (1983) pp. 13-17; Riedel (1988); McCann (1988); Bidgoli (1989) pp. 27-34; Zachary (1988) pp. 997-1030
11.1.6.1.b	Pew (1988) pp. 931-940
11.1.6.1.c	Zachary (1988) pp. 997-1030
11.1.6.1.d	Zachary (1988) pp. 997-1030
11.1.6.1.e	Riedel (1988)
11.1.6.2	Zachary (1988) pp. 997-1030
11.1.6.2.a	Mittal (1985) pp. 32-36
11.1.6.2.b	Zachary (1988) pp. 997-1030
11.1.6.2.c	Tannenbaum (1990) pp. 54-59; Riedel (1988); O'Keefe (1989) pp. 217-226; Ma et al. (1989) pp. 996-1012
11.1.6.2.d	Finke and Lloyd (1988) pp. 170-193; Wairath (1989)
11.1.6.3	Barnett (1990) pp. 1552-1556
11.1.6.4	Ehrhart (1990)
11.1.6.4.a	LeMay (1988) pp. 227-229

REFERENCES (Cont.)

<u>Paragraph</u>	<u>Reference</u>
11.1.6.4.b	Pew (1988) pp. 931-940
11.1.6.4.c	Schwartz (1983) pp. 13-17
11.1.6.5	Schwartz (1983) pp. 13-17
11.1.6.5.a	Gordon (1988) pp. 55-59; Schwartz (1983) pp. 13-17
11.1.6.5.b	Tannenbaum (1990) pp. 54-59
11.2	Hansen and Hansen (1992); Flynn (1987) p. 221; Harter (1986) pp. 124-127; Frost (1984) p. 8; Humphrey and Melloni (1986) pp. 41-50; Kelley (1984); Katzeff (1986); Martin (1983) pp. 427-483; Hershman Kelly and Miller (1979); Schur (1988); Shneiderman in Vassiliou (1984); Boehm-Davis (1989); Ogden and Brooks (1983)
11.2.1	Martin (1983) pp. 427-483
11.2.1.1	Tenopir and Lundeen (1988) pp. 43-45
11.2.1.2	Frost (1984) pp. 187; Martin (1983) pp. 143, 452
11.2.1.3	Martin (1983) pp. 452
11.2.1.4	Grill (1990) pp. 78
11.2.1.5	Cuff (1980)
11.2.1.7	Boehm-Davis (1989)
11.2.1.8	Boehm-Davis (1989)
11.2.1.9	Boehm-Davis (1989)
11.2.1.10	Flynn (1987) pp. 401-441
11.2.2	Galitz (1989) pp. 155-165; Flynn (1987) pp. 401-441
11.2.2.1e	Tenopir and Lundeen (1988) p. 45
11.2.2.5f	Boehm-Davis (1989)
11.2.3	Humphrey and Melloni (1986) pp. 200-203
11.2.3.1a	Chapnick (1989)
11.2.3.3d	Humphrey and Melloni (1986) pp. 143-144, 200; Tenopir and Lundeen (1988) pp. 34; Ehrenreich (1982)
11.2.3.5	Harter (1986) pp. 154
11.2.3.5a	Shneiderman in Vassiliou (1984) p. 5
11.2.4	Tenopir and Lundeen (1988) pp. 44-45; Humphrey and Melloni (1986) pp. 197-199
11.2.4.1d	Flynn (1987) pp. 520-521
11.2.4.1e	Flynn (1987) pp. 514-515
11.2.4.1f	Benbasat and Wand (1984); Carlson and Metz (1980)
11.2.4.4	Tenopir and Lundeen (1988) pp. 155-167
11.2.4.5	Frost (1984) pp. 241-242; Feldman and Rogers (1982)
11.2.5	Harter (1986) pp. 76-94

REFERENCES (Cont.)

<u>Paragraph</u>	<u>Reference</u>
11.2.5.1	Harter (1986) p. 28
11.2.5.2	Lochovsky and Tsichritzis in Vassiliou (1984) p. 131
11.2.5.4	Schauer in Blaser and Zoeppritz (1983) p. 33
11.2.5.5	Humphrey and Melloni (1986) p. 200; Tenopir and Lundeen (1988) p. 34; Ehrenreich (1982)
11.2.5.5c	Grill (1990) p. 78
11.2.5.6	Tenopir and Lundeen (1988) pp. 31-39
11.2.5.6a	Frost (1984) pp. 196-197
11.2.5.6c	Sormunen in Wormell (1987)
11.2.5.6j	Humphrey and Melloni (1986) p. 200; Harter (1986) pp. 41-58; Ogden and Brooks (1983)
11.2.5.6k	Humphrey and Melloni (1986) p. 202
11.2.5.6n	Sormunen in Wormell (1987)
11.2.6	Humphrey and Melloni (1986) pp. 199-200; Benbasat and Wand (1984); Tenopir and Lundeen (1988) p. 45; Shneiderman in Vassiliou (1984) pp. 4, 7; Harter (1986) Schur (1988); Kelley (1984); Flynn (1987) pp. 504-506; Cuff (1980); Sormunen in Wormell (1987)
11.3.1.1	Nicol (1990) p. 115
11.3.1.2	Kearsley (1988) p. 27; Shneiderman (1987) pp. 371-372; Carroll and Mazur (1986) p. 38
11.3.1.3	Shneiderman (1987) p. 374
11.3.1.4	Kearsley (1988) p. 99
11.3.1.5	Gery (1991) p. 59
11.3.1.6	Kearsley (1988) p. 22
11.3.1.7	Wexelblat (1989) p. 74
11.3.1.8	Helander (1990) p. 603
11.3.1.8a	Helander (1990) p. 603
11.3.1.9	Roth et al. (1988) p. 915
11.3.1.10	Getler (1991) pp.14-22
11.3.1.11	Roth et al. (1988) p. 98; Carroll and McKendree (1987) p. 24
11.3.1.12	Avery (1992) Personal Communication
11.3.1.13	Knerr (1992) Personal Communication
11.3.1.14	Kearsley (1988) p. 77
11.3.1.15	Fernandes and Maracle (1991) p. 9; Ripley (1989) pp. 811-822
11.3.1.16	Roth et al. (1988) p.98

REFERENCES (Cont.)

<u>Paragraph</u>	<u>Reference</u>
11.3.2.1	Kearsley (1988) p. 66, 81; Wexelblat (1989) p. 76
11.3.2.1a	Helander (1990) p. 353
11.3.2.1b	Kearsley (1988) p. 66, 81
11.3.2.1c	Kearsley (1988) p. 66, 81
11.3.2.2	Carroll and McKendree (1987) p. 23
11.3.3.1	Sellen and Nichol (1990) p. 146; Kearsely (1988) p. 77
11.3.3.2	Gery (1991) p. 59
11.3.3.3	Gery (1991) p. 49
11.3.3.4	Gery (1991) p. 59
11.3.3.4a	Wexelblat (1989) p. 76
11.3.3.4b	Carroll and McKendree (1987) p. 25
11.3.3.4c	Carroll and McKendree (1987) p. 25
11.3.3.4d	Carol and Mazur (1986) p. 39
11.3.3.5	Gery (1991) p. 59; Wenger (1987) p. 124
11.3.3.6	Gery (1991) p. 137
11.3.3.7a	Seybold's Office Computing Report (1989) p. 6
11.3.3.7b	Wexelblat (1989) p. 76
11.3.3.7c	Wexelblat (1989) p. 76
11.3.3.8	Wenger (1987) p. 80-86; MIL-STD 1379D (12.5.1990) p.5
11.3.4.1	Wenger (1987) p. 336
11.3.4.2	Fernandes and Maracle(1991) p. 9
11.3.4.3	Fernandes and Maracle (1991) p. 9; Seybold's Office Computing Report (4.1989) p.6
11.3.5.1	Badler and Webber (1991) p.71; (1990) p. 638
11.3.5.2	Helander (1990) p. 358
11.3.5.3	Nicol (1990) p. 115; Roth et al. (1988) p. 118
11.3.5.3a	Carol and Mazur (1986) p. 38
11.3.5.3b	Roth et al. (1988) p. 118; MIL-STD 1379D (12.5.1990) p. C-5
11.3.5.3c	Roth et al. (1988) p. 118
11.3.5.4	Carol and Mazur (1986) p. 37; Walker (1987) pp. 238-243
11.3.5.5	Kearsley (1988) p. 24
11.3.5.6	Roth et al. (1988) p. 30
11.3.5.7	Wexelblat (1989) p. 74
11.3.5.8	O'Malley et al. (1983) p.6; Carroll (1982) pp. 49-58

REFERENCES (Cont.)

<u>Paragraph</u>	<u>Reference</u>
11.3.6.1	Kearsley (1988) p. 67
11.3.6.2	Seybold's Office Computing Report (4.1989) p. 5
11.3.6.3	Kearsley (1988) p. 79
11.3.7.1	Fernandes and Maracle (1991) p. 9
11.3.7.2	Brooks et al. (1990) p. 1387
11.3.7.3	Fernandes and Maracle (1991) p. 9
11.3.7.4	Kearsley (1988) p. 76
11.3.8	Sellen and Nicol (1990) p. 145; Helander (1990) p. 360
11.3.9.1	Fernandes and Maracle (1991) p.10
11.3.9.2	Fernandes and Maracle (1991) p. 9
11.3.10.1	Roth et al. (1988) p. 36
11.3.10.1a	Roth et al. (1988) p. 36
11.3.10.1b	Wenger (1987) pp. 296-297
11.3.10.2	Roth et al. (1988) p.36
11.3.10.3	Roth et al. (1988) p.36
11.3.10.4	MIL-STD-1472D (1981) p.274; Shneiderman (1987) p.317; Smith and Mosier (1986) p.391
11.3.10.5a	Wexelblat (1989) p.76
11.3.10.5b	Wexelblat (1989) p.76
11.3.10.5c	Carroll and McKendree (1987) p. 24
11.3.10.5d	Carroll and McKendree (1987) p. 24
11.3.10.5e	Kearsley (1988) pp. 20, 80
11.3.10.5f	Carol and Mazur (1986) p. 40
11.3.10.5g	Carroll and McKendree (1987) p. 24
11.3.11.1	Kearsley (1988) p.23
11.3.11.2	Kearsley (1988) p.23
11.3.11.3	Wenger (1987) p. 319
11.3.11.4	Wenger (1987) p. 319
11.3.11.5	Gery (1991) p.63

APPENDIX A

SECURITY PRESENTATION GUIDELINES

This appendix seeks to provide a uniform HCI across all Compartmented Model Workstation (CMW) applications.

The security portion of the HCI will comply with DDS-2600-6216-89 and the DIA Style Guide, from which this appendix is derived. These documents outline security-related interface requirements for workstations operating in the System High or Compartmented Mode. To ensure consistency, however, any DoD workstation security label displayed by an application should conform to the labeling guidelines in this appendix.

(Note: Although the figures in this appendix are only drawn in the Motif style, the security relevant information is common to both Motif and Open Look applications.)

A.1 LABEL STANDARDIZATION

The following guidelines for label presentation apply to all CMWs:

A.1.1 GUIDELINES FOR LABEL SYNTAX

One of the primary display requirements is to use the long names of words in all labels.

A.1.1.1 Sensitivity Labels

1) The syntax for output of sensitivity labels for all CMWs is as follows:

CLASSIFICATION COMPARTMENT COMPARTMENT COMPARTMENT....

Examples: TS A
 TS A B C D

2) The syntax for input of sensitivity labels for all CMWs is as follows:

CLASSIFICATION COMPARTMENT COMPARTMENT COMPARTMENT....
CLASSIFICATION/COMPARTMENT/COMPARTMENT/COMPARTMENT....

A.1.1.2 Information Labels

- 1) The syntax for output of information labels to the security banner on all CMWs is as follows:

CL CW M REL C

where CL is the classification, CW is zero or more blank-separated codewords, M is zero or more blank-separated non-codeword markings, and C is either a single country code or multiple country code separated by slashes. The long form of classification, code-words, and markings are used for output. The short form of classification, codewords, and markings may be used otherwise (e.g., list file command). Code words, markings, and "REL" country codes are displayed in the order in which the words appear in the encodings file.

Example: TOP SECRET BRAVO1
 SECRET ALPHA1 NOFORN
 TOP SECRET B SB REL UK
 SECRET ORCON ORG X REL UK/CAN/AUS

- 2) The syntax for input of information labels on all CMWs is as follows:

CL CW M REL C

where CL is the classification, CW is zero or more blank-separated codewords, M is zero or more blank-separated non-codeword markings, and C is either a single country code or multiple country codes separated by slashes. The short or long form of classification and markings may be used for input. When entering an information label, codewords, markings, and "REL" country codes may be entered in any order.

Example: TS B1
 TOP SECRET B1
 S A1 NF
 SECRET A1 NF
 TS B1 REL UK
 TOP SECRET A SA REL UK
 S OC OX REL UK/CAN/AUS
 SECRET OC OX REL UK/CAN/AUS

A.1.1.3 Information and Sensitivity Labels Together

- 1) The syntax for output of sensitivity labels and information labels together (as in the Classification Bar of each base window) is as follows:

CL CW M REL C [CL COMPARTMENT COMPARTMENT...]

where CL is the classification, CW is zero or more blank-separated codewords, M is zero or more blank-separated markings, and C is either a single country code or multiple country codes separated by slashes.

The information labels are always followed by two or more blanks, followed by the sensitivity label enclosed in square brackets. In sensitivity labels, the short form of the classification is used. In information labels, the long form of the classification is used. The long form of the codewords and/or markings is used in the information label. If the information and sensitivity labels are to be output to other than the Classification or Input Information Labels (e.g., list files with labels), then the short form of classification, codewords, markings, etc. may be used.

Example: TOP SECRET A B SA [TS A B]
SECRET A B SA SB [TS A B C D]
TOP SECRET A B ORCON ORG X REL CAN/UK [TS A B]

- 2) At times when both labels can be input, the user must enter the left bracket to delimit the sensitivity label. The short or long form of the codeword and/or markings may be used for input of both labels.

Example: TS A B SA [TS A B]
TS A1 B1 [TS A B]
S A B PX LD (TS A B C D)
S A B PROJECT X LIMIDIS [TS A B C D]
TS A SA OC OX REL CAN/UK [TS A B C]
TS A B ORCON ORG X REL CAN/UK [TS A B C]

A.1.2 GUIDELINES FOR DISPLAYING LABELS

The following guidelines apply for all displaying of information labels, sensitivity labels, and clearance labels.

- 1) Capital letters should be displayed for all classifications and words in all labels.
- 2) Blanks should be used to separate classifications from other words in all labels, except where there are multiple words that require the same prefix or suffix, in which case the multiple words should be separated from each other with slashes.
- 3) The long name of words should be displayed in all labels.
- 4) The long name of classifications should be displayed in all information labels.
- 5) The short name of classifications should be displayed in all sensitivity labels and clearances.
- 6) The classification should be displayed first, followed by the words in the same order they appear in the encodings.
- 7) Whenever an information label is displayed, its associated sensitivity label should also be displayed.

A.1.3 GUIDELINES FOR CHANGING LABELS

A.1.3.1 Typing Interface

The following guidelines apply to all textual interfaces that allow users to change labels:

- 1) When typing any label or a change to any label, the user should be able to use the following interchangeably:
 - Upper and lower case letters
 - Short and long names for classifications and words
 - Blanks and slashes

- 2) The following syntax should be accepted for typed changes to any label:

[+][CLASSIFICATION] [[+]-][WORD]...

where brackets denote optional entries, "+" or "-" denotes changes, and "... " denotes zero or more of the previous bracketed entry preceded by blanks. If the input starts with a classification followed by "+" or "-", the new classification should be used but the rest of the old label should be retained and modified as specified in the input.

- 3) It should never be possible for the user to change the value of an information label above that of its associated sensitivity label without first, or concurrently, requesting that the sensitivity label be raised appropriately.
- 4) The following syntax should be accepted when the user can change both the information and associated sensitivity labels of an object:
- New information label or changes - when changes are only to information label
 - [New sensitivity label or changes] - when changes are confined to sensitivity label
 - New information label or changes [New sensitivity label or changes] - when there are changes to both information and sensitivity labels, or when user wants to set sensitivity label to same level as information label
- 5) The user should be shown the label resulting from the changes and asked to confirm them before they are finally made by the system.
- 6) The existence of classifications and words for which the user is not cleared should be hidden from the user by treating such classifications or words in a manner identical to classifications or words that are not defined in the encodings.
- 7) Whenever a user enters multiple hierarchically related words in the same label, only the highest of the words should remain in the label.

- 8) Whenever a user enters multiple words that cannot be combined in the same label, only the first of the words specified should remain in the label.
- 9) To the maximum extent unambiguously possible, errors made in typing changes to labels should be corrected, with error messages to the user.

A.1.3.2 Graphical User Interface (GUI)

The following guidelines apply to GUIs that allow label changing via the selection of individual classifications and words. Whenever reference is made in this section to a mouse, other similar pointing devices (e.g., trackballs) are also acceptable.

- 1) The graphical interface should be integrated with the typing interface, such that the user can specify changes using either the mouse or by typing.
- 2) A character string representation of the label should be visible after each mouse selection.
- 3) Only classifications and words that are valid for the user to select for a label or are required in the label should be displayed in association with that label.
- 4) Each classification or word should be annotated to indicate whether or not the classification or word is present in the label.
- 5) Each classification or word should be separately annotated if it cannot always be selected.
- 6) Each change specifiable through typing should have an analogy in the GUI.
- 7) When displayed as selections in the GUI, classifications should be visually separated from words, words should appear in the order specified in the encodings, and the first pure marking word should be visually separated from the previous words. These separations should be accomplished without identification of the various components on the display (e.g., classifications, compartments, markings) because there is no universally accepted identification terminology.

- 8) If all potential selections cannot fit on the screen, a scrolling or paging mechanism should be implemented to display all selections.

A.2 WINDOW STANDARDIZATION

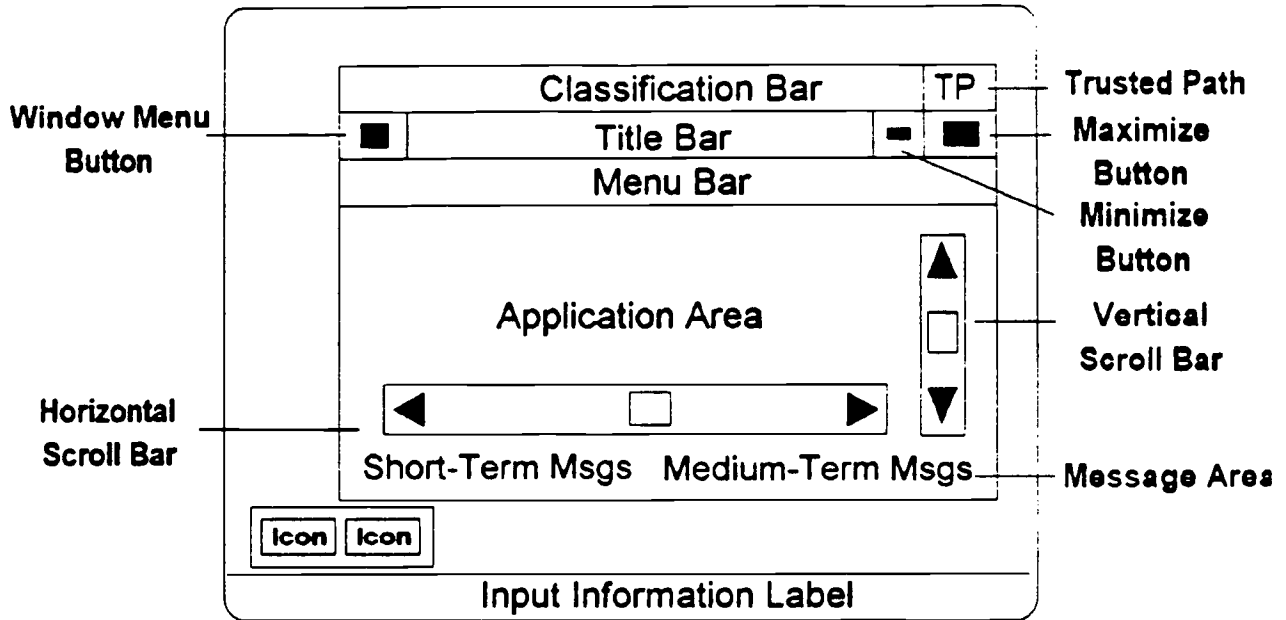
Window standardization is necessary to avoid confusion when users move from one CMW to another and to simplify training. To provide a standard user interface, CMW vendors will comply with the following guidance:

A.2.1 SCREEN PRESENTATION

- 1) When the machine is turned on (and before CMW is booted), the user will be presented with a distinctive screen (Trusted Path) either through color, screen marking, font, or combination thereof. The user authorized to boot the CMW will be presented with a prompt to enter an appropriate user-id and, if validated, password (see figure A.1). After booting, a user log-in screen will be displayed.
- 2) The CMW will be configurable so at least three windows can be displayed upon user authentication.
- 3) Icons will default to the lower left portion of the screen, starting from the lower left corner and moving right as more icons are created.

A.2.2 CLASSIFICATION BAR

- 1) A Classification Bar for output of sensitivity and information labels will be applied to each application base window created by the CMW. The Classification Bar will appear directly above the Title Bar.
- 2) A Trusted Path button will be placed in the Classification Bar at the far right of the window. The Trusted Path button will be distinguishable by using the same mechanism (screen marking, color, font, or combination thereof) used for the Trusted Path at initial log-in.
- 3) If the information label, spacing separator, sensitivity label, and Trusted Path button are longer than the space provided in the Classification Bar, the window manager will provide the following default for trimming the labels:



Open Look

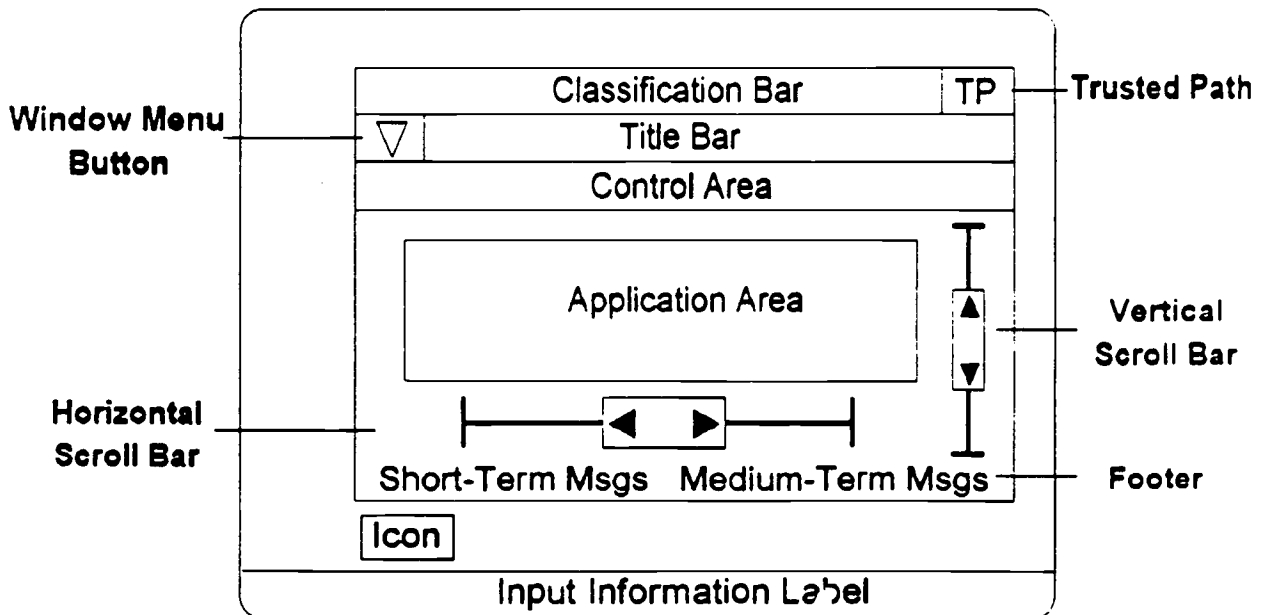


FIGURE A.1: Sample CMW Screens

- Labels will be truncated from the right beginning with the information label and followed by the sensitivity label.
- Truncation of the information label will be denoted by a dash followed by a greater than symbol "->" on the left side of the label. Truncation of the information label will stop when only one character of the label remains.
- Truncation of the sensitivity label will be denoted by a less than symbol followed by a dash "<-" on the right side of the label. Truncation of the sensitivity label will stop when only one character of the label remains, not including the left square bracket ([).

Example: TOP SECRET A B SA SB NOFORN [TS A B]
->TOP SECRET A B SA SB [TS A B]
->TOP SECRET [TS A B]
->TOP [TS A B]
->T [TS A <-
->T [T<-

- 4) The user must be able to view the entire information label by positioning the pointer in the classification bar and pressing any mouse button, or by menu selection via the Trusted Path.

A.2.3 RESERVED AREA OF THE SCREEN

- 1) One of the CMW Trusted Path requirements is that the CMW will "provide reserved portions of the screen to which user processes cannot write." Normally, the Input Information Label will be displayed in the reserved area. Additionally, a visual indicator (e.g., blinking or highlighted Trusted Path symbol) should be displayed in this area when the Trusted Path is active (e.g., when a window classification is being changed). The Trusted path menu can be invoked by moving the pointer to the Reserved Area and depressing the Motif Custom or Open Look Menu mouse button.
- 2) If the Input Information Label is too long for the space provided, the window manager will provide the following default for trimming the label:
 - Input Information Labels will be truncated from the right.

- Truncation of the Input Information Label will be denoted by a dash followed by a greater than symbol "->" on the left side of the label.

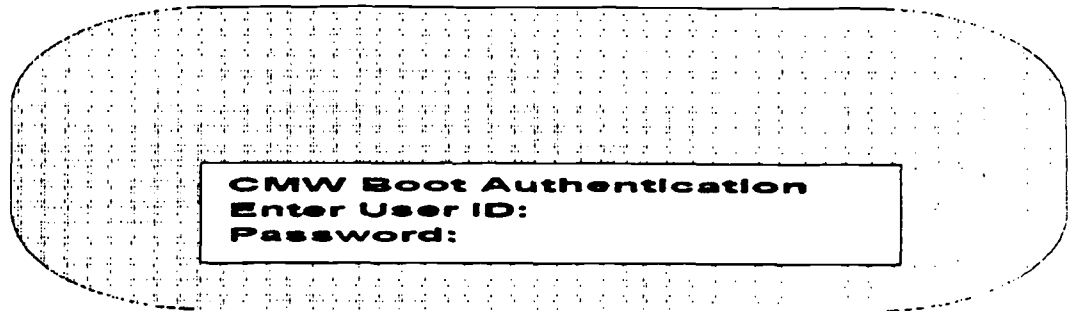
Example: TOP SECRET A B SA SB
->TOP SECRET A B SA SB PROJECT X/Y LIMDIS ORCON O

- 3) The user must be able to view the entire Input Information Label by positioning the pointer in the reserved area and pressing the select mouse button or through menu selection via the Trusted Path menu. Also, upon selecting the appropriate menu item, the user may change the Input Information Label for the active window using a dialog box.

A.2.4 TRUSTED PATH BUTTON

- 1) When the user clicks on the Trusted Path button, a pop-up menu with the following minimum options will be displayed in the center of the screen (see figure A.2):
 - Create New Window (specifying the sensitivity level and input level or letting it default to the current window)
 - Change Password
 - Change Information Label of a file (not of the current window)
 - Change Input Information Label
 - User Authorization ->
- 2) When a user selects "User Authorization->," a cascading menu will be displayed. The contents of the menu will depend on the individual's USER-ID (whether a "normal" user, Information System Security Officer (ISSO), Systems Administrator, or Operator), and the authorizations given that individual from the Trusted Facility Management. If the user clicks outside the bounds of the user authorization menu, the display will return to the Trusted Path main menu.
- 3) When an authorized normal user (one whom the ISSO has included in the Access Control List of the privileged program) selects "User Authorization->," a cascading menu with the following minimum

BOOT SCREEN



ACTIVE SCREEN

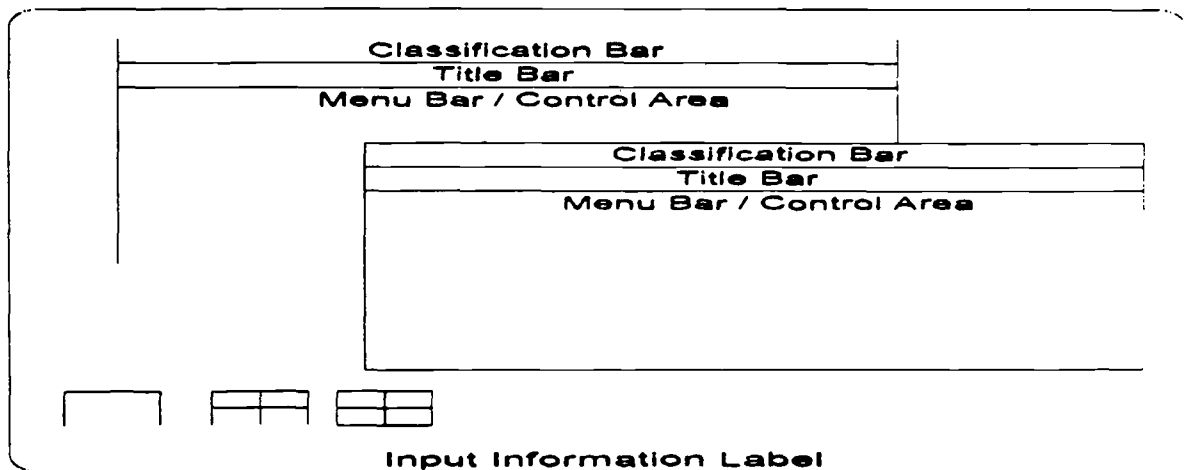


FIGURE A.2: Sample Trusted Path Main Menu

authorizations will be displayed:

- No Classification Marking - allows a user to bypass the requirement for printing information labels on the top and bottom of each page of printed output. (It does NOT permit the user to bypass the requirement for a print banner at the beginning and end of the output.)
- Change Sensitivity Label - allows a user to change the sensitivity label of a file the user owned.

- **Set Sensitivity = Information Label** - allows a user to set the sensitivity label of a file the user owns to the information label.
- 4) When the ISSO selects "User Authorization->," a cascading menu with the following minimum options will be displayed:
- **Change Sensitivity Label** - allows the ISSO to set the sensitivity label of any file (file, device, file system, etc.) on the CMW.
 - **Assign User Password/Clearance** - allows the ISSO to set the password, password length, password time-limit, and clearance of a user's profile.
 - **File Privileges** - allows the ISSO to set/show the privileges associated with files.
 - **Operator Administrator Authorization** - allows the ISSO to set/show the authorizations for the other users, the administrator, and operators.
 - **Set Audit Events** - allows the ISSO to configure the auditing system.
 - **Review Audit Data** - allows the ISSO to reduce and review the audit data and store the audit data on removable media.
- 5) When the System Administrator selects "User Authorization->," a cascading menu with the following minimum options will be displayed:
- **Add User** - allows the administrator to create a user profile, assign the user to a group, and create a user home directory. The ISSO must then assign the user a password.
 - **Add Group** - allows the administrator to create a discretionary access group.
 - **ISSO Authorization** - allows the administrator to set the authorizations of the ISSO.
- 6) When the operator selects "User Authorization->," a cascading menu with the following minimum options will be displayed:

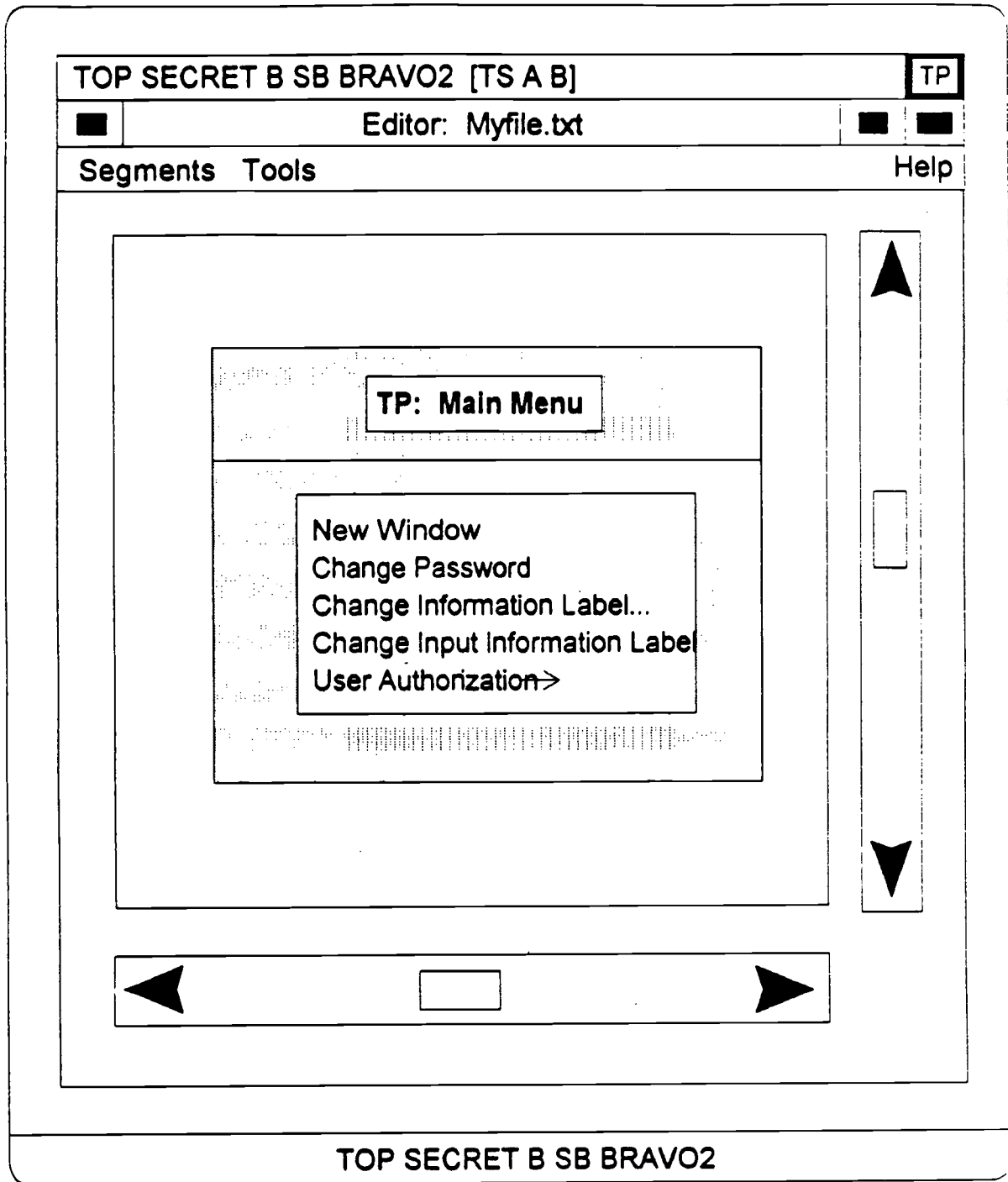


FIGURE A.3: Sample for Entering New Input Information Label

- **Make File System Backups** - allows the operator to back up file system on the CMW.
- **Configure Printer** - allows the operator to configure the printer (maximum sensitivity, printer definition, communication port, etc.).
- **Enable Printer** - allows the operator to enable printer service to users.
- **Disable Printer** - allows the operator to disable a printer for reconfiguration, etc.
- **Mount a file System** - allows the operator to mount a file system on the CMW.
- **Unmount a File System** - allows the operator to unmount a file system from the CMW.
- **Halt System** - allows the operator to halt the CMW.

A.3 CLASSIFICATION DISPLAY ENHANCEMENTS

This guide designates a minimum security functionality, which the vendor is encouraged to supplement. An example could be to provide a menu box from the Trusted Path that builds an information label for a specified file (see figure A.3). Clicking a mouse button over the classification, multiple codewords, multiple handling caveats, and multiple release markings could dynamically build the appropriate information label.

Another example could be to implement Trusted Path differently depending on whether it was invoked using the Trusted Path button or through the Trusted Path Background. The menu from the Trusted Path button could present options related to windows (Create, Cut/Paste, etc.) and the menu from the background selection could present non-window related items (Change Password, User Authorizations, etc.).

APPENDIX B - GLOSSARY

AC: See alternating current.

accelerator key: Special key or key combination that performs the same action as a menu selection.

Ada: High-level computer programming language developed by the Department of Defense (DoD). Ada is used as the standard programming language for DoD. It is used for real-time processing, is modular in nature, and includes object-oriented features.

alternating current (AC): Electrical current that reverses its direction at regularly recurring intervals.

API: Application Programming Interface

application: Classification of computer programs designed to perform specific tasks, such as word processing, database management, or graphics.

applications menu: List of options within an application.

automated tools: Software performing a sequence of operations to assist the user in achieving a goal (e.g., within graphics software, functions that align objects, smooth curves, or draw circles).

backlighting: Lit from the back. When referring to a monitor, light passes through the display screen from the back in order to illuminate screen images.

base-level functions: Initial or basic functions.

batch processing: Processing data or the accomplishment of jobs accumulated in advance in such a manner that each accumulation thus formed is processed or accomplished in the same computer run.

baud: Measure of the transmission speed capability of a communications line or system. In a sequence of binary signals, the rate of one baud equals one bit per second.

bit-mapped display: Display in which every picture element (pixel) of the screen can be referenced individually.

bookmarking: Method of tagging items of interest to the user for easy referral later. Allows the user to customize the application.

Boolean logic: Logical expression that uses Boolean operators such as AND, OR, NOT, XOR, NOR, and NAND to create a statement that, when resolved, is either true or false.

Boolean operators: A keyword in programming that causes two values to be combined in a logical fashion.

branching menu: Menu which, if selected, brings up another menu.

bring-to-front: Process of moving a window to the foreground.

Candela (cd): Unit of luminous intensity expressed in Candela per square meter (cd/m^2). One cd is equal to 0.29 footLambert.

CAP: Computer/Electronic Accommodation Program.

Cathode ray tube: Electronic vacuum tube that focuses electrons energizing phosphors on a screen, creating a visible display. The typical computer monitor uses this type of display technology.

cd: See candela.

central processor: Portion of the computer that controls execution of applications.

character: Single letter, digit, or symbol.

character string: Series of alphanumeric characters, the contents of which are treated as though they were text.

CMW: Compartmented Mode Workstation.

COBOL: Acronym for Common Business-Oriented Language. COBOL is a computer programming language used extensively in mainframes and minicomputers for business applications.

command: Entry that instructs the computer to effect a specific action.

command entry: Informing the computer that a specific command should be effected.

command language: Limited programming language used strictly for executing a series of commands.

command stacking: Allows the user to key a sequence of commands as a single "stacked" command entry.

compatible letters: Letters easily associated with the function requested, for example, "P" for print, "Q" for quit.

context-sensitive: Computer action or response directly related to the cursor position or specific point in the software, for example, a help function that displays information about the specific data entry field in which the cursor was located when help was called.

control action: Actions which must be effected to control a window or other graphics object or its contents.

control entry: Input action by the computer user that changes some aspect of the appearance or function of the application.

control lockout: Processing delay that results in pacing the capability to enter sequences of control commands.

COTS: Commercial-Off-The-Shelf software.

courseware: Another name for educational or training materials and software.

crosstalk: Optical crosstalk, or bleeding, occurs when the light from the incorrect video image gets through. When referring to stereoscopic images, the right eye's image is visible to the left eye or vice versa.

CRT: See Cathode Ray Tube.

cursor: Visual mechanism to mark, on-screen, where current input or output is to happen.

data entry: Series of keystrokes used to input information into the computer.

data entry field: Space (number of characters and/or digits) allowed for data entry.

data field: Location in a file or database that contains a specific type of information.

database: Structured or organized collection of information, which may be accessed by the computer.

database management system: Computer application program that accesses or manipulates the database.

DC: See direct current.

default: Command which is automatically executed if none is specifically indicated.

default value: Value of a variable in lieu of a specifically indicated value.

defeated: Option that cannot be selected due to another selected option's use.

delimiter: Symbols such as commas, spaces, or parentheses which mark the boundaries of a specific block of information.

designate: Process of selecting and displaying the current or active window with visual cues.

destructive entries: Any entry which will destroy or overwrite information.

DIA: Defense Intelligence Agency.

dialog. Structured series of interchanges between a user and a computer terminal. Dialogs can be initiated by the computer or the user. Interactive dialog consists of an action by the user followed by a response from the computer or vice versa.

dialog box: Screen display box containing a message requesting additional information from the user.

direct current (DC): Electrical current that flows in one direction only and is substantially constant in value.

direct manipulation: Method of data organization (typically involving extensive windowing and iconization) in which the user can select specific displays of information and move them about to facilitate interaction with an application. A system of interaction in which the user's actions directly affect software operations.

display frame: Window or page.

display parallax: When used in discussing touch screen technology, display parallax is the apparent displacement of an object viewed on a curved CRT screen and seen through a flat touch interactive display.

display screen: Screen of a multipage file.

DoD: U.S. Department of Defense.

DODIIS: Department of Defense Intelligence Information Systems.

double keying: Each character of the data item does not have an appropriately labeled key and therefore requires more than one keystroke for entry.

dual activation: Two keys are used simultaneously to input a command.

dynamic depth displays: Stereoscopic displays that are designed to change (move) images during viewing.

electroluminescence (EL): Luminescence produced by electrical excitation of phosphor in powder or film form.

electronic mail: Communication, processed through a network, from one workstation to another.

end user: Person who ultimately uses the computer application or output.

error management: Various options within an application that allow the user to eliminate the effects of commands executed accidentally or unwisely.

expand: Ability to resize objects to produce better organization of on-screen material, usually a graphic or a window.

fc: See footcandle.

feedback: Visual acknowledgement that the computer is executing the command or that the command was executed.

field: Addressable data location.

file: Any specifically identified collection of information stored in the computer.

FIPS: Federal Information Processing Standard.

Fl: See footLambert.

footcandle (fc): Unit of measurement of illumination. The amount of light emitted by a standard candle (1 cd) measured one foot away from the candle equals one footcandle.

footLambert (Fl): Unit of measure of intensity of reflected or emitted light (luminance). The average luminescence of any reflecting surface in footLamberts is the product of the illumination in footcandles by the luminous reflectance of the surface.

frame: Single display image or screen.

function key, fixed and variable: Key which, when depressed, effects a specific action. It can either be a single, predefined function (fixed), or vary according to the system mode or level within an interactive dialog.

form filling: Method of interaction in which the user enters a series of commands or data items in predefined fields. These fields may be mandatory or optional.

FORTRAN: Acronym for FORMula TRANslator, which is a high level computer language used extensively in scientific and engineering applications.

freeze: See Option - PAUSE.

GOTS: Government-Off-The-Shelf software.

graphical interaction: Transactions between the user and computer-generated graphical representations of objects (screens, menus, buttons, etc.).

Graphical User Interface (GUI): System design that allows the user to effect

commands, enter into transaction sequences, and receive displayed information through graphical representations of objects (menus, screens, buttons, etc.).

GUI: See Graphical User Interface.

hard copy: Printed copy of machine output in a visibly readable form. For example, printed reports, listings, documents, summaries.

hardware architecture: Assemblage of a computer's internal components and its attached peripheral devices which determine its capabilities and its limitations.

hatching: Graphical pattern characterized by 45 and 135 degree diagonal lines which cross the patterned area.

HCI: See Human Computer Interface.

help screen: Separate window that offers advice and information on how to overcome a specific problem and/or to better interact with the computer.

HFE: See Human Factors Engineering.

hierarchical menu: Method of organizing menus in layers. The secondary or tertiary menus are stored within a primary menu.

high level language: Programming language that does not reflect the structure of any one computer or class of computers.

high resolution: Screen display within an extremely fine visual reproduction of detail.

highlight: Visual method to call attention to a specific piece of text or a graphic through differentiating it from surrounding texts or graphics. This is usually accomplished using contrasting colors or reverse video.

hook: Selecting a corner of a window or icon in order to move or resize it.

Human-Computer Interface (HCI): Hardware and software allowing information exchange between the user and the computer.

Human Factors Engineering (HFE): Approach that makes use of scientific facts in the design of items (i.e. computer systems, software, etc.) to produce effective human-machine integration and utilization.

icon: Graphical representation of an object, concept, or message used by a computer system to represent items such as files, documents, programs, and disk drives.

Iconify: Process that changes the text representation of an object, concept, or message into an icon.

Iconification: Process of iconifying.

IEEE: Institute of Electrical and Electronics Engineers.

illuminance: Measure of the quantity (density) of light reaching an object or surface. Measured in footcandles.

Image Formation Time (IFT): Measurement of the time required to update screen image displays.

Infrared (IR): Radiation outside the visible light range on the red side (wavelength 0.75 to 0.8 micrometers).

Input focus: Applies to a window that actually receives user input. This window is known as the active window where keyboard input appears and pointing device inputs apply. "Explicit" input focus refers to user or application action (e.g., typing keyboard accelerators, clicking pointer inside a window, moving a window through menu selection, etc.) to assign input focus. "Implicit" focus refers to focus automatically assigned to the window containing the location cursor.

interactive control: Attribute describing the ability of a program and a user to interface with each other during program execution.

Interactive dialogue: See dialog.

Interactive procedures: Methods by which a user interacts with a computer and the computer with the user.

Interface: Interconnection and interrelationships between two devices, two applications, or the user and an application or device.

Interlock: Mechanism to connect two or more processes within a computing system to ensure that no one part of a hardware or software system can be operated independently.

interocular: Perceptual coordination between the eyes.

IFT: See image formation time.

IR: See infrared.

jump-ahead: Capability of moving ahead during a step-wise process to allow quicker performance of an operation; useful for experienced computer users.

justification: Alignment of text on a display or a printed page. Left justification means that the left margin is even.

keyword: Special word in a programming language that tells the computer which operation to perform.

Lambert: See footLambert.

landscape: Screen display or printing orientation parallel to the wide side of the paper.

LCD: See Liquid Crystal Display.

LCSS: See Liquid Crystal Stereoscopic Shutter.

left-justified: See Justification.

Liquid Crystal Display (LCD): Display operated by polarizing light in which the nonactive segment reflects incident light and thus appears invisible against its background.

Liquid Crystal Stereoscopic Shutter (LCSS): Type of display that utilizes liquid crystal shutters, one for each eye synchronized to alternate fields of the display, and representing one of the two images necessary to achieve the third dimension.

lockout: Condition of the application locking the keyboard (i.e., not accepting commands from it) while the application is executing a command.

log on: Process of gaining access to the system, usually involving a password and a recognition of the specific user by the computer.

logarithm: The exponent that indicates the power to which a number has been raised to produce the given number.

$$N = 10^n \quad \log_{10} N = n$$

luminance: Amount of light per unit area reflected from or emitted by a surface. Measured in footcandles.

lux: Standard measure of illuminance. One lux is one lumen per square meter.

macro: Executable file that stores a series of commands and keystrokes to be used later.

MANpower and PeRsonnel INTEgration (MANPRINT): An Army program that addresses concerns with manpower, personnel, training, human factors, system safety, and health hazards.

MANPRINT: Acronym; MANpower and PeRsonnel INTEgration.

masking: Partial or complete obscuring of one item by another.

memory: Place in the computer in which information is stored.

menu: List of options available within a software application.

menu bar: The horizontal menu usually at the top of the screen which contains menu titles.

metaphor: System-level analogy used for the grouping of processes and/or procedures. Usually associated with icons based on the analogy. As, for example, a desk top metaphor where icons represent office equipment or operations.

minimize: Procedure to make the window as small as it can be without being closed; this is usually done through iconization.

mnemonic: Word or code symbolic of another word, code, or function.

mode: Status of the screen or program process.

Modulation Transfer Function (MTF): A parameter using spatial frequency responses to characterize a screen display. The spatial frequency is stated in lines (line pairs) or minimum/maximum intensity pairs per unit distance. The MTF is used as a performance measurement of many optical systems.

Motif: User interface design approach based upon the "look" and "feel" presented in the OSF/Motif style guide. Motif is marketed by the Open Software Foundation.

MTF: See Modulation Transfer Function.

multifunction keying: Interface design where computer keys may perform multiple functions with the use of a combination of keystrokes.

multindow: Simultaneous display of several windows on the computer screen.

natural language: Programming language paradigm exemplified by using English-like commands and syntax to issue commands; interactions in the vernacular of the user.

navigation: Manner in which the user moves through the menu structure.

NATO Forces: Personnel in the military forces of member nations of the North Atlantic Treaty Organization (NATO).

NIST: National Institute of Standards and Technology.

nit: See normalized intensity.

normalized intensity (nit): Metric unit of measure of luminous intensity. A nit is equal to one candela per square meter (cd/m²) or 0.29 footLambert.

null: Empty; nothing. A null set contains no elements.

OCR: See Optical Character Recognition.

one to many mapping: An icon which represents a category of possibilities within an option is a one to many mapping.

one to one mapping: An icon which represents a single, specific function is a one to one mapping.

Open Software Foundation (OSF): Consortium of computer hardware and software manufacturers whose membership includes over seventy of the computer industry's leading companies.

open window map: A map (graphic display) that shows windows that are open and how they relate to each other.

open: Procedure to cause a window to be displayed from an icon or menu option so a document, directory or file can be viewed.

Optical Character Recognition (OCR): The analysis and translation of a graphic representation of text into a coded form such as ASCII or EBCDIC.

option: Command that may be selected to access a specific function of an application.

option - BACKUP: Option that will display the last transaction or the process of saving information to non-volatile memory.

option - CANCEL: Command that allows the user to have the computer disregard a previous command.

option - CONFIRM: Explicit warning of any possible data loss.

option - CONTINUE: Option that resumes a transaction sequence which has been stopped by a PAUSE.

option - GOBACK: Option that will display the last transaction. See also BACKUP.

option - PAUSE: Option that temporarily causes a transaction sequence to stop running. Use the CONTINUE option to resume after pausing.

option - RESTART: Option that will cancel any entries that have been made in a transaction sequence and return to the beginning of a sequence.

option - REVIEW: Option that returns to the first display of a transaction sequence, allowing the user to review the transaction and make necessary changes.

option - SUSPEND: Option that allows a user to leave the application, then, when he returns, resume at the same point he left off.

option - UNDO: Option that immediately reverses any action.

option code: Codes associated with the available choices.

OSF: See Open Software Foundation.

output: Information the computer displays in response to the user's actions.

overarching guidelines: Dominant or all-embracing guidelines.

overlapping: Windowing system in which one window covers a portion of another.

overlay: Printing or drawing on a transparent or semi-transparent medium on the same scale as a map, chart, etc., to show details not appearing or requiring special emphasis on the original.

paging: Scrolling through material one page at a time.

paired opposites: Set of opposite functions such as up and down, top and bottom.

pan: Process to change the displayed region (often of a map) in a regular and smooth manner.

parallax: Apparent displacement of an object as seen from two different points not on a straight line with the object.

parameter: Quantity or constant whose value varies with the circumstances of the application.

piezoelectric: Electric polarity due to pressure especially in a crystalline substance.

pixel: Contraction for picture element. A pixel is a single dot on a display screen.

pixel matrix: Arrangement of screen dots (pixels) to form text or graphic displays.

pop-up menu: Lists of options that appear on the display screen in the form of a window.

portrait: Screen or printing orientation parallel to the narrow side of the paper.

predictive modelling: Use of a model to predict the actual response of a system or process.

preformatted: Screen structure prepared for the user.

presentation graphics: Pictorial representations of the relationships between variables (graphs and charts) or representations of systems (diagrams, schematics, and graphical renditions).

primitive: Code that defines a specific elementary shape, form, or color.

programming language: Artificial language established for expressing computer programs.

prompt: Text or graphic display that indicates the start point for user-generated actions. This term is also used for software generated instructions for process confirmation.

pull-down menu: Lists of options attached to a selection on a menu bar.

push-to-back: Process of moving a window to the background.

query language: Specialized type of command language to elicit information from the computer system.

real time: Absence of delay, except for the time required for transmission.

real-time control system: Systems capable of responding to external events with negligible delays.

resize: Procedure to change the size of a window or graphic.

resize border: Border of a window that, if selected, will allow the user to resize the window.

resolution: Density of picture display elements of the screen; degree of detail with which an image is displayed or printed.

retrieve: Procedure required to display stored information for purposes of viewing and manipulation.

right-justified: See justification.

SAW: See Surface Acoustic Wave.

scroll: Method used to move the contents of a window or list in a dialogue box using the scroll bar or scroll arrows.

scroll bar: Rectangular bar that may be along the right edge or bottom of a window. Clicking or dragging in the scroll bar causes the view of the document to change.

secondary coding: Providing more than one method for coding displayed information. For example, in coding a particular item with red color, the use of the symbol "R" would provide secondary method for conveying the information when color was not available.

semantics: Relationship of characters or groups of characters to their meanings, independent of the manner of their interpretation and use.

sensitivity analysis: Study that shows the response of a system to varying conditions. For example, "How sensitive is the system to increased workload?"

sequence control: Prescribed control over the order of function performed by the computer; this impacts the way in which a user interacts with the application.

size coding: Variations in the size of displayed alphanumerics and symbols. Such coding can be used for categorization.

slider: Part of the scroll bar which indicates what part of the file contained in a window is being viewed.

soft keys: Visual representation of key functions on the display screen. This is usually associated with software controlled function key capabilities.

specular reflector: Reflecting light in a diffuse manner.

stacked command: Single command composed of multiple commands that must be executed individually.

stereopsis: Phenomenon of simultaneous vision with two eyes in which there is a vivid perception of distance of objects from the viewer (three dimensional or stereoscopic vision).

stereoscopic: Method of seeing objects in three dimensions.

stroke width (sw): Width of the line used to create a displayed character.

subordinate window: A window that is opened from and controlled by another window.

subtend: Opposite in position.

summary symbols: Symbol which categorizes the information portrayed by a group of symbols.

supraordinate window: Higher level window, usually the window from which subordinated options or tasks are controlled.

Surface Acoustic Wave (SAW): when used in the context of touch screen technology, an approach that uses ultrasonic sound "beams" transmitted from two perpendicular sides of a display frame.

sw: See stroke width.

system level menu: List of which applications are available for utilization.

system response time: Amount of time that elapses between a command being given and its being executed by the computer.

text editor: Application that allows text to be created or modified.

text-based systems: Method of organization in which the primary form of interaction between the system and user is through text rather than through graphical or voice interaction.

three-dimensional: Relating to the three physical dimensions (height, width, depth). Giving the effect of depth or varying distances.

TID: See Touch Interactive Display.

tiling: Windowing approach in which multiple windows do not overlap, rather, all lie on the same plane.

theater-type displays: Display screens suitable for large group presentations as used in a movie theater or auditorium.

title banner: Horizontal bar at the top of a window which shows the name of the window and allows it to be moved.

Touch Interactive Display (TID): Uses a physical device between the user and the display which acts as the input mechanism.

transaction: Interaction between a user and a computer in which the user inputs a command to receive a specific result from the computer.

transaction sequence: Order of transactions required to accomplish the desired results.

transmissivity: Measurement of the ability of an image to be transmitted. When used in the context of touch screen technology, refers to the ability of the image to be transmitted through a filter placed in front of a computer screen.

type-ahead: Capability of the computer to receive commands faster than it can display their results.

UCI: See User-Computer Interface.

UIDL: User Interface Definition Language.

UIMS: User Interface Management System.

user-callable: Able to be requested by the user as desired.

User-Computer Interface (UCI): Hardware and software allowing information exchange between the user and the computer.

user-specified windows: Windows whose content has been selected by the user.

variable: Quantity that can assume any of a given set of values.

VDT: See Video Display Terminal.

Video Display Terminal (VDT): Terminal composed of a keyboard for data input and a CRT screen for display of the input/output.

widget: Basic graphical object which is a component of a user interface component.

window: Typically rectangular display which provides a visual means for interaction with an application.

zoom: Graphical tool used to magnify a portion of a document for more detailed viewing.

APPENDIX C - REFERENCES

Abramson, S. R., L. H. Mason, and H. L. Snyder. 1983. "The Effects of Display Errors and Font Styles Upon Operator Performance with a Plasma Panel." In Proceedings of the Human Factors Society - 27th Annual Meeting, pp. 28-32. HFS, Santa Monica, California.

ACM - see Association for Computing Machinery, Inc.

Air Force Intelligence Agency. 1990. Air Force Intelligence Data Handling System Style Guide. Air Force Intelligence Agency, Washington, D.C.

Ambron, S., and K. Hooper. 1990. Learning with Interactive Multimedia: Developing and Using Multimedia Tools in Education. Microsoft Press, Redmond, Washington.

Andrews, J. R., W. E. Haas, and M. D. Rainsdon. 1989. "Holographic Displays for the Man-Machine Interface." Optical Engineering, 28(6):643-649.

Andriole, S. J., and L. A. Adelman. 1990. "Prospects for Cognitive Systems Engineering." Presentation, Center for Multidisciplinary Information Systems Engineering, Drexel University, Philadelphia.

Antin, J. 1988. "An Empirical Comparison of Menu Selection, Command Entry and Combined Modes of Computer Control." Behavior and Information Technology 7(2) 173:173-182.

Armstrong, H. G. 1988. "Colored Light-Emitting Diodes: Use of Red or Green in High Ambient Illumination." In Engineering Data Compendium: Human Perception and Performance, eds. K. R. Boff and J. E. Lincoln, Vol. 3:2260-2261. Harry G. Armstrong Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio.

Aspillaga, M. 1991. "Screen Design: Location of Information and Its Effects on Learning." Journal of Computer-Based Instruction 18(3):89-92.

Association for Computing Machinery, Inc. (ACM). 1989. Human Factors in Computing Systems: CHI '89 Conference Proceedings, Austin, Texas. ACM Press, New York.

Association for Computing Machinery, Inc. (ACM). 1990. Resources in Human-

Computer Interaction. ACM Press, New York.

Auerbach Publishers, Inc., ed. 1981. Practical Data Base Management. Reston Publishing Company, Inc., Reston, Virginia.

Avery, L. W., R. V. Badalamente, S. E. Bowser, P. A. O'Mara, S. E. Reynolds. 1990. Human Factors Design Guidelines for the Army Tactical Command and Control System (ATCCS) Soldier-Machine Interface, Version 1.0. Pacific Northwest Laboratory (PNL) for the U.S. Army Tactical Command and Control System (ATCCS) Experimentation Site (AES), Fort Lewis, Washington.

Avery, L. W., S. E. Bowser, S. M. Adams, R. V. Badalamente, D. T. Donohoo, D. A. Gellert, K. D. Hargrove, W. A. Hesser, J. G. Heubach, P. A. O'Mara, D. J. Pond, R. B. Randall, S. E. Reynolds, and M. S. Rowley. 1992. Human Factors Design Guidelines for the Army Tactical Command and Control System (ATCCS) Soldier-Machine Interface, Version 2.0, eds, L. W. Avery and S. E. Bowser. Pacific Northwest Laboratory (PNL) for the U.S. Army Tactical Command and Control System (ATCCS) Experimentation Site (AES), Fort Lewis, Washington.

Badler, N., and B. Webber. 1991. "Task Communication Through Natural Language and Graphics." AI Magazine 11(5):71-73.

Baeker, R. 1980. "Towards an Effective Characterization of Graphical Interaction." IFIP Workshop on Methodology of Interaction, pp. 127-147.

Baggen, E. A., H. L. Snyder, and M. R. Miller. 1988. "A Human Factors Evaluation of Current Touch-Entry Technologies." In SID International Symposium Digest of Technical Papers, pp. 259-262. SID, Playa Del Rey, Kingsbeach, California.

Bailey, R. W. 1982. Human Performance Engineering: A Guide for System Designers. Prentice Hall, Englewood Cliffs, New Jersey.

Bailey, R. W. 1989. Human Performance Engineering: Using Human Factors/Ergonomics to Achieve Computer system Usability. Prentice Hall, Englewood Cliffs, New Jersey.

Baker, C. 1988. Text-Editing Performance as a Function of Screen Size: A Pilot Study. 612716.H7000700011, U.S. Army Human Engineering Laboratory, Aberdeen Proving Ground, Maryland.

Barnett, B. J. 1990. "Aiding Type and Format Compatibility for Decision Aid Interface

Design." In Proceedings of the Human Factors Society 34th Annual Meeting, pp. 1552-1556. HFS, Santa Monica, California.

Barten, P. G. J. 1991. "Resolution of Liquid-Crystal Displays." In Society for Information Display 91 Digest, pp. 772-775. SID, New York.

Beaton, R. J. 1990. "Human Factors Engineering of 3D Stereoscopic Display Systems." In Stereographics, 17th International Conference on Computer Graphics and Interactive Techniques, Association for Computing Machinery, Special Interest Group on Graphics, Dallas, Texas.

Beaton, R. J., and S. T. Knox. 1987. "Flat Panel Image Quality." In Society for Information Display 87 Digest, pp. 115-118. SID, New York.

Benbasat, I., and Y. Wand. 1984. "A Structured Approach to Designing Human-Computer Dialogues." International Journal of Man-Machine Studies 21(2):105-126.

Benson, B. L., and J. E. Farrell. 1988. "The Effect of Character Height-to-Width Ratio on CRT Display Legibility." In Society for Information Display 88 Digest, pp. 340-343. SID, New York.

Biberman, L. M., and B. Tsou. 1991. Image Display Technology and Problems with Emphasis on Airborne Systems. Technical Report No. AD-B1157 161, Defense Technical Information Center, Alexandria, Virginia.

Bidgoli, H. 1989. "DSS Products Evaluation: An Integrated Framework." Journal of Systems Management 40(11):27-28

Billingsley, P. A. 1988. "Taking Panes: Issues in the Design of Windowing Systems." In Handbook of Human-Computer Interaction, ed. M. Helander, pp. 413-434. Elsevier Science Publishers B.V., Amsterdam.

Blaha, R. J. 1992. Emerging Large Screen Display Technology. Technical Report No. M92B0000010, MITRE Bedford, Massachusetts.

Blaha, R. J. 1990. Large-Screen Display Technology Assessment for Military Applications. Technical Report No. M90-16, MITRE Bedford, Massachusetts.

- Blaha, R. J., C. D. Crotty, and D. F. Martin. 1992. Evaluation of Large-Screen Display Systems in Workstation Development Environment. Working Paper No. 92B0000081, The MITRE Corp., Bedford, Massachusetts.
- Blaser, A., and M. Zoeppriz, eds. 1983. Enduser Systems and Their Human Factors. Springer-Verlag, Berlin.
- Blattner, M. M., D. A. Sumikawa, and R. M. Greenberg. 1989. "Earcons and Icons: Their Structure and Common Design Principles." Human-Computer Interaction 4:11-44.
- Bobko, D. J., P. Bobko, and M. A. Davis. 1986. "Effect of Visual Display Scale on Duration Estimates." Human Factors Society, 28(2):153-158. HFS, Santa Monica, California.
- Boehm-Davis, D. A., R. W. Holt, M. Koll, G. Yastrop, and R. Peters. 1989. "Effects of Different Data Base Formats on Information Retrieval." Human Factors 31(5):579-592.
- Bosman, D. 1991. "Simulation of the Perception of a Computer-Generated Display Image." Proceedings of Society for Information Display, 32(1):3-12. SID, New York.
- Bowser, S. E. 1991. "Review of Army Tactical Command and Control System Soldier-Machine Interface Functional Issues." Pacific Northwest Laboratory (PNL) Task Order 91-04, Subtask 2b for the U.S. Army Tactical Command and Control System (ATCCS) Experimentation Site (AES), Fort Lewis, Washington.
- Breen, P. T. 1987. "Functional Requirements for C3I Large Screen Displays." In Large Screen Projection Displays, 760:2-8. SPIE Press, Bellingham, Washington.
- Breen, P. T. 1989. Functional Requirements for C3I Large Screen Displays. Technical Report No. M89-01, MITRE Bedford, Massachusetts.
- Breen, P. T., and H. C. Masterman. 1990. Modulation Depth Metrics for the Large Screen Displays in NASA's Mission Control Centers. Working Paper No. 28736, The MITRE Corp. Bedford, Massachusetts.
- Breen, P. T., P. E. Miller-Jacobs, and H. H. Miller-Jacobs. 1987. "Color Displays Applied to Command, Control, and Communications (C3) Systems." In Color and the Computer, ed. J. H. Durrett, pp.171-187. Academic Press, Inc., Boston.
- Brooks, J. D., R. D. Gilson, and H. R. Myler. 1990. "Display Design Guide for Bivisual

Media." Proceedings of the Human Factors Society 34th Annual Meeting 1990. HFS, Santa Monica, California.

Brosda, V., and G. Vossen. 1988. "Update and Retrieval in a Relational Database Through a Universal Schema." ACM Transactions on Database Systems 13(4):449-485.

Brosley, M., and B. Shneiderman. 1978. "Two Experimental Comparisons of Relational and Hierarchical Database Models." International Journal of Man-Machine Studies 10:625-637.

Brown, C. M. 1989. Human-Computer Interface Design Guidelines. Ablex Publishing Corporation, Norwood, New Jersey.

Brown, C. M., D. B. Brown, H. V. Burkleo, J. E. Mangelsdorf, R. A. Olsen, and R. D. Perkins. 1983. Human Factors Engineering Standard for Information Processing Systems. LMSC-D-877141 (Revision of C410), Lockheed Missiles and Space Company, Inc., Los Angeles.

Bullinger, H. J., K. P. Fährnich, and J. Ziegler. 1987. "Software-Ergonomics: History, State-of-the-Art, and Important Trends." In Cognitive Engineering in the Design of Human-Computer Interaction and Expert Systems, ed. G. Salvendy, pp. 307-316. Elsevier Science Publishers B.V., Amsterdam.

Campbell, J. A., and S. P. Ross. 1987. "Issues in Computer-Assisted Interpretation of Graphs and Quantitative Information." In Cognitive Engineering in the Design of Human-Computer Interaction and Expert Systems, ed. G. Salvendy, pp. 473-480. Elsevier Science Publishers B.V., Amsterdam.

Carbone, R. M., and D. MacIver. 1987. "A Survey of Large-Screen Displays of C3I Applications." In Large Screen Projection Displays, 760:6-10. SPIE Press, Bellingham, Washington.

Carey, J. M., ed. 1988. Human Factors in Management Information Systems. Ablex Publishing Corporation, Norwood, New Jersey.

Carlson, E. D., and W. Metz. 1980. "Integrating Dialog Management and Data Base Management." Information Processing: Proceedings of the IFIP Congress, Vol. 80, pp. 463-468. North-Holland Publishing Co., Amsterdam.

Carroll, J. M., and S. A. Mazur. 1986. "LisaLearning." Computer 19(11):35-49.

Carroll, J. M., and J. McKendree. 1987. "Interface Design Issues for Advice-Giving Expert Systems." Communications of the ACM 30(1):14-31.

41 CFR. Federal Information Resources Management Regulations (FIRMR), Chapter 201. Government Printing Office, Washington, D.C.

Chao, B. P. 1987. "Prototyping a Dialogue Interface: A Case Study." In Cognitive Engineering in the Design of Human-Computer Interaction and Expert Systems, ed. G. Salvendy, pp. 357-363. Elsevier Science Publishers B.V., Amsterdam.

Chao, B. P. 1986. Design Guidelines for Human-Computer Dialogues. SAND86-0259, Sandia National Laboratories, Albuquerque, New Mexico.

Chapnick, P. 1989. "Ushering in a New Era." AI Expert, Feb. 1989, p. 7-8, Miller Freeman, Inc.

Christensen, C., E. A. Baggen, and H. L. Snyder. 1985. "Performance Measures and Subjective Evaluations for Two Color Displays." In Proceedings of the Human Factors Society - 29th Annual Meeting, pp. 1075-1078. Santa Monica, California.

Clarke, A. A. 1986. "A Three-Level Human-Computer Interface Model." International Journal of Man-Machine Studies 24:503-517.

Cowen, M. 1991. A Comparison of Four Types of Feedback During Computer-Based Training (CBT). NPRDC-TR-92-2, Navy Personnel Research and Development Center, San Diego, California.

Crolotte, A., J. Saleh, and A. Freedy. 1980. "Human Decision Processes in Command and Control of Marine Amphibious Brigade Operations, IEEE, pp. 1216-1220.

Cuff, R. N. 1980. "On Casual Users." International Journal of Man-Machine Studies 12(2):163-187.

Davies, S. P., A. J. Lambert, and J. M. Findlay. 1989. "The Effects of the Availability of Menu Information During Command Learning in a Word Processing Application." Behavior and Information Technology 8:135-144.

Decker, J. J., C. J. Dye, C. J. C. Lloyd, and H. L. Snyder. 1991. The Effects of Display Failures and Symbol Rotation on Visual Search and Recognition Performance. Technical Memorandum 4-91. U.S. Army Human Engineering Laboratory, Aberdeen

Proving Ground, Maryland.

Decker, J. J., P. L. Kelly, K. Kurokawa, and H. L. Snyder. 1991. The Effect of Character Size, Modulation, Polarity, and Font on Reading and Search Performance in Matrix-Addressable Displays. Technical Memorandum 6-91. U.S. Army Human Engineering Laboratory, Aberdeen Proving Ground, Maryland.

Defense Information Systems Agency/Center for Information Management (DISA/CIM). 1992a. Comparison of the IEEE Recommended Practice for Graphical User Interface Drivability with the DISA Center for Information Management Human Computer Interface Style Guide. Technical Memorandum, DISA/CIM, Washington, D.C.

Defense Information Systems Agency/Center for Information Management (DISA/CIM). 1992b. Human Computer Interface Style Guide, Version 1.0. DISA/CIM, Washington, D.C.

Defense Information Systems Agency/Center for Information Management (DISA/CIM). 1991. Technical Reference Model for Corporate Information Management (Draft). DISA/CIM, Washington, D.C.

Defense Intelligence Agency (DIA). 1990. "DIA Standard Military Graphics Symbols Manual" (DIAM 65-xx) (Draft). DIA, Washington, D.C.

Defense Intelligence Agency (DIA). 1989. Compartmented Mode Workstation Labeling: Source Code and User Interface Guidelines. DIA Document DDS-2600-6215-89, DIA, Washington, D.C.

Defense Intelligence Agency (DIA). 1983. DIA Standard User Interface Style Guide for Compartmented Mode Workstations. DIA Memorandum U-15, 284/DSE-3, Washington, D.C.

Department of Defense Intelligence Information Systems (DODIIS) Management Board. 1991. DoD Intelligence Information Systems Style Guide. DoD, Washington, D.C.

Doll, T. J. 1991. "Electro-Optic/Infrared Technology and the Human-Machine Interface." In Proceedings of the Human Factors Society - 35th Annual Meeting, pp. 1500-1501. HFS, Santa Monica, California.

Dominessy, M. E. 1989. A Literature Review and Assessment of Touch Interactive Devices. Technical Memorandum 11-89. U.S. Army Human Engineering Laboratory, Aberdeen Proving Ground, Maryland.

Dumas, J. S. 1988. Designing User Interfaces for Software. Prentice Hall, Englewood Cliffs, New Jersey.

Durrett, H. J., ed. 1987. Color and the Computer. Academic Press, Inc., Boston.

Dye, C. J., and H. L. Snyder. 1991. The Effects of Display Failure, Polarity, and Clutter on Visual Search for Symbols on Cartographic Images. Technical Memorandum 9-91, U.S. Army Human Engineering Laboratory, Aberdeen Proving Ground, Maryland.

Ehrenreich, S. L. 1981. "Query Languages: Design Recommendations Derived from the Human Factors Literature." Human Factors 23(6):709-725.

Ehrhart, L. S. 1990. "New Approaches to Human-Computer Interaction Research and Design for Decision Aiding Systems." In Proceedings of the 5th IEEE International Symposium on Intelligent Control. IEEE Computer Society Press, Los Alamitas, California.

Elkerton, J., R. C. Williges, J. A. Pittman, and J. Roach. 1982. "Strategies of Interactive File Search" in Proceedings of the Human Factors Society. HFS, Santa Monica, California.

Farooq, M. U., and W. D. Dominick. 1988. "A Survey of Formal Tools and Models for Developing User Interfaces." International Journal of Man-Machine Studies 29:479-496.

Feldman, M. B., and G. T. Rogers. 1982. "Toward the Design and Development of Style-Independent Interactive Systems." In Human Factors in Computer Systems: Proceedings, March 15-17, 1982, pp. 111-116. ACM, New York.

Fenchel, R. 1981. "An Integrated Approach to User Assistance." ACM SIGSOC Bulletin 13 2:98-104.

Fernandes, K. 1992. User Interface Specifications for Navy Command and Control Systems, Version 1.1. U.S. Department of the Navy; Naval Command, Control, and Ocean Surveillance Center, Research, Development, Test, and Evaluation Division, San Diego, California.

Fernandes, K., and K. E. Maracle. 1991. Design Standards for the Command and Control Information Navigation and Training System (CINTS). Technical Note 1660, Naval Ocean Systems Center, San Diego, California.

Finke, D. J., and M. M. Lloyd. 1988. "Guidelines and Tools for Human-Decision Aid Interface Design and Evaluation." In Proceedings of the Fifth Annual Workshop on Command and Control Decision Aiding, pp. 170-193, September 1988.

FIPS - See National Institute of Standards and Technology (NIST)

Flynn, R. R. 1987. An Introduction to Information Science. Marcel Dekker, Inc., New York.

Frost, R. A. 1984. Database Management Systems. McGraw-Hill, New York.

Funk, H. L. 1989. "Information Display - An Overview and Trends." In Proceedings of the Institution of Electrical and Electronics Engineers, pp. 2-1 - 2-7. IEEE, Washington, D.C.

Gaines, B. R., and M. L. G. Shaw. 1986. "Foundation of Dialog Engineering: The Development of Human-Computer Interaction." Part II. International Journal of Man-Machine Studies 24:101-123.

Gaines, B. R., P. V. Facey. 1975. "Some Experience in Interactive System Development and Application." In Proceedings of the IEEE 63(6):894-911.

Galitz, W. O. 1989. Handbook of Screen Format Design. QED Information Sciences, Inc., Wellesley, Massachusetts.

Galitz, W. O. 1984. Humanizing Office Automation. QED Information Sciences, Inc., Wellesley, Massachusetts.

Garner, K. H. 1990. "20 Rules for Arranging Text on a Screen." CBT Directions:13-16.

General Services Administration (GSA). 1991. Managing Information Resources for Accessibility. GSA, Clearinghouse on Computer Accommodation (COCA) of the Information Resources Management Service (IRMS), Washington D.C.

Gery, G. 1991. Electronic Performance Support Systems. Weingarten Publications, Boston, Massachusetts.

Getler, R. 1991. "The Case for Concurrent Authoring." CBT Directions:14-22.

Gilmore, W. E., D. I. Gertman, and H. S. Blackman. 1989. "The User-Computer Interface in Process Control." In A Human Factors Engineering Handbook, pp. 21-33. EG&G Idaho, Inc., Idaho Falls, Idaho.

Gittins, D. 1986. "Icon-Based Human-Computer Interaction." International Journal of Man-Machine Studies 24:519-543.

Glasser, J., and A. Rolland. 1989. "Visual Performance Evaluation for LCD Displays: Appropriate Methods for Measuring Luminance and Contrast." Proceedings of Society for Information Display, 1077:9-20. SID, New York.

Gold, R. S., and A. G. Ledebuhr. 1985. "Full Color Liquid Crystal Light Valve Projector." In Advanced in Display Technology V, 526:51-58. SPIE Press, Bellingham, Washington.

Goode, W. F. 1991. "Status of Electronic Displays." Seminar for the Society of Information Display, SID, New York.

Gordon, S. E. 1988. "The Human Factor in Expert Systems," AI Expert, pp. 55-59.

Grabinger, R. S., and D. Amedeo. 1988. "CRT Text Layout: Perceptions of Viewers." Computers in Human Behavior 4:189-205.

Greitzer, F. 1987. Comments on draft NASA Guidelines Document: User/Computer Interaction Section. Memo on 1, 22, 1987.

Grether, W. F., and C. A. Baker. 1972. Human Engineering Guide to Equipment Design, eds., H. P. Van Cott and R. C. Kinkade. GPO, Washington, DC.

Grill, E. 1990. Relational Databases: A Methodical Guide for Practical Design and Implementation. Ellis Harwood, New York.

GSA - See General Services Administration

Gunderson, J., G. Gruetzmacher, and N. Swanson. 1991. "Legibility of Seven Segment Numeric LED Displays: Comparisons of Two Fonts at Various Distances." In Proceedings of the Human Factors Society - 35th Annual Meeting, pp. 491-495. HFS, Santa Monica, California.

Hamel, C. J., and S. L. Clark. 1986. CAI Evaluation for the Design of Computer-Aided Instruction. Technical Report NTSC TR86-002 (DTIC No. AD-A172383). Naval Training Systems Center, Orlando, Florida.

Hannigan, S., and V. Herring. 1987. "Human Factors in Office Product Design - European Practice." In Proceedings of the Second International Conference on Human-Computer Interaction, Conference Title "Cognitive Engineering in the Design of Human-Computer Interaction and Expert Systems," ed. G. Salvendy, pp. 226-232. Elsevier Science Publishers B.V., Amsterdam.

Hansen, G. W., and J. V. Hansen. 1992. Database Management and Design. Prentice Hall, Englewood Cliffs, New Jersey.

Harrell, T. H. 1987. "Designing User-Computer Dialogues: Basic Principles and Guidelines." In Computerized Psychological Testing: Current Issues and Future Directions, Honaker, L. M., Chair. Symposium conducted at the American Psychological Association, New York.

Harter, S. P. 1986. On-line Information Retrieval. Academic Press, Orlando, Florida.

Hawkrigde, D. G. 1988. Computers in Company Training. Croom Helm, London.

Heines, J. M. 1984. Screen Design Strategies for Computer-Assisted Instruction. Digital Press, Bedford, Massachusetts.

Helander, M., ed. 1988. Handbook of Human-Computer Interaction. Elsevier Science Publishers B.V., Amsterdam.

HFS - See Human Factors Society

Hildreth, C. R. 1982. Online Public Access Catalogs: The User Interface. Online Computer Library Center, Inc. Dublin, Ohio.

Hoadley, E. D. 1990. "Investigating the Effects of Color." Communications of the

ACM 33(2):120-127.

Holtzman, S. 1989. Intelligent Decision Systems. Addison-Wesley, Reading, Massachusetts.

Horton, W. K. 1990. Designing and Writing Online Documentation; Help Files to Hypertext. John Wiley and Sons, New York.

Human Factors Society, Inc. (HFS). 1988. American National Standard for Human Factors Engineering of Visual Display Terminal Workstations. ANSI/HFS 100-1988, HFS, Santa Monica, California.

Humphrey, S. M., and B. J. Melloni. 1986. Databases: A Primer for Retrieving Information by Computer. Prentice Hall, Englewood Cliffs, New Jersey.

Hunter, M. W. 1988. "CRT Anti-glare Treatments, Image Quality, and Human Performance." Ph.D. Dissertation, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

Hurd, J. C. 1983. "Writing Online Help." In Proceedings of 30th International Technical Communication Conference, pp. 151-154. Society for Technical Communication, Washington, D.C.

Hurley, A. F., and L. I. Hoffberg. 1991. Evaluation of Six Large-Screen Displays. Technical Report No. 29399, MITRE Bedford, Massachusetts.

Hussein, B. 1989. "DSS Products Evaluation: An Integrated Framework." Journal of Systems Management, pp. 27-34, November 1989.

Hyland, R. M., and P. R. Boivin. 1987. Performance Characteristics of Infrared Touch Screens for Typical Workstation Applications. Technical Report No. NSU 8093, Defense Technical Information Center, Alexandria, Virginia.

IBM. 1984. Human Factors Workstations with Visual Displays, Third Edition. San Jose, California.

Inmon, W. H. 1981. Effective Data Base Design. Prentice Hall, Englewood Cliffs, New Jersey.

Institute of Electrical and Electronics Engineers (IEEE). 1992a. "Draft Standard for Information Technology-Uniform Application Program Interface-Graphical User Interfaces," IEEE P1201.1 Draft 3.1. Institute of Electrical and Electronics Engineers Standards Department, Piscataway, New Jersey.

Institute of Electrical and Electronics Engineers (IEEE). 1992b. "IEEE Recommended Practice for Graphical User Interface Drivability," IEEE P1201.2, Draft 2. Institute of Electrical and Electronics Engineers Standards Department, Piscataway, New Jersey.

International Standards Organization (ISO). 1991. "Part 3 Flat Panel Addendum, Additional Notes for Reviewers." In "ISO 9241, Working Draft of ISO Committee 159 SC 4 Working Group 2." ISO 9241, ISO, Washington, D.C.

ISO (see International Standards Organization)

Jarvenpaa, S. L., and G. W. Dickson. 1988. "Graphics and Managerial Decision-Making: Research Based Guidelines." Communications of the ACM 31(6):764-775.

Katzeff, C. 1988. "The Effect of Different Conceptual Models Upon Reasoning in a Database Query Writing Task." International Journal of Man-Machine Studies 29(1):37-62.

Katzeff, C. 1986. "Dealing With a Database Query Language in a New Situation." International Journal of Man-Machine Studies 5(1):1-17.

Kawamura, T., H. Kawamura, K. Kobara, A. Saitoh, and Y. Endo. 1991. "Anti-reflection Coating for Inner Surface of CRT Faceplate." In Society for Information Display 91 Digest, pp. 49-52. SID, New York.

Kearsley, G. 1988. Online Help Systems: Design and Implementation. Ablex Publishing, Norwood, New Jersey.

Kelley, J. F. 1984. "An Iterative Design Methodology for User-Friendly Natural Language Office Information Applications." ACM Transactions on Office Information Systems 2(1):26-41.

Kelster, R. S., and G. R. Galloway. 1983. "Making Software User Friendly: An Assessment of Data Entry Performance." Proceedings of the Human Factors Society. HFS, Santa Monica, California.

Kieras, D. E., and S. Boviar. 1984. "The Role of a Mental Model in Learning to

Operate a Device." Cognitive Science 8:255-273.

Klien, G. A., and D. MacGregor. 1988. "Knowledge Elicitation of Recognition-Primed Decision-Making." Technical Report 799. U.S. Army Research Institute Field Office, Fort Leavenworth, Kansas.

Korth, H. F., A. Silbershatz. 1986. Database System Concepts. McGraw-Hill, New York.

Krause, J. 1980. "Natural Language Access to Information Systems: An Evaluation Study of Its Acceptance by End Users." Information Systems 5(4):297-318.

Kubota, H., T. Marushige, T. Takabayashi, and S. Kobayashi. 1986. "LCD Legibility." In Society for Information Display 88 Digest, pp. 157-160. SID, New York.

Kubota, H., T. Marushige, C. M. Gomes, and S. Kobayashi. 1988. "Legibility of Multiplexed Dot-Matrix LCDS: The Effect of Surface Reflection." In Proceedings of Society for Information Display (SID), 29(3):213-216. SID, New York.

Lansdale, M. W. 1988. "On the Memorability of Icons in an Information Retrieval Task." Behavior and Information Technology 7:131-151.

Lanzetta, T. M., and N. D. Lubart. 1988. "Comparing the Readability of Display Technologies: Paper, CRT, and LCD." In Society for Information Display 88 Digest, pp. 336-339. SID, New York.

Larson, H. T. 1989. "Large Automated Flat Situation Display for the Commander." Maintaining Start-of-the Art in ACCS:3-44. Army Science Board Summer Study.

Laurel, B., ed. 1990. The Art of Human Computer Interface Design. Addison-Wesley, Reading, Massachusetts.

Laverson, A., K. Norman, and B. Shneiderman. 1987. "An Evaluation of Jump-Ahead Techniques in Menu Selection." Behavior and Information Technology 6(2):97-108.

- Laycock, J. 1985. "The Legibility of Passive Displays". Proceedings of Society for Information Display, 26(2):89-93. SID, New York.
- LeMay, M. 1988. "Why Some Decision Aids Work and Others Do Not." Proceedings of the 1988 IEEE International Conference on Systems, Man, and Cybernetics, pp. 227-229.
- Lewis, H. V., and J. J. Fallesen. 1989. Human Factors Guidelines for Command and Control Systems: Battlefield and Decision Graphics Guidelines. Research Project 89-01. U.S. Army Research Institute for the Behavioral and Social Sciences, Alexandria, Virginia.
- Lickteig, C. W. 1989. Design Guidelines and Functional Specifications for Simulation of the Battlefield Management Systems (BMS) User Interface. U.S. Army Research Institute for the Behavioral and Social Sciences, Alexandria, Virginia.
- Lloyd, C. J. C., J. J. Decker, and H. L. Snyder. 1991. The Effects of Line and Cell Failures on Reading and Search Performance Using Matrix-Addressable Displays. Technical Memorandum 7-91. U.S. Army Human Engineering Laboratory, Aberdeen Proving Ground, Maryland.
- Lloyd, C. J. C., J. J. Decker, K. Kurokawa, and H. L. Snyder. 1988. "Effects of Line and Cell Failures on Reading and Search Performance Using Matrix-Addressable Displays." In Society for Information Display 88 Digest, pp.344-346. SID, New York.
- Lochovsky, F. H, and D. C. Tschritzis. 1984. "Querying External Databases." In Human Factors and Interactive Computer Systems, ed. Y. Vassiliou. Ablex Publishing Corporation, Norwood, New Jersey.
- Lysaght, R. J., R. Harris, and W. Kelly. 1988. "Artificial Intelligence for Command and Control." Technical Report 2122. U.S. Army Communications and Electronics Command, Fort Monmouth, New Jersey.
- Ma, P., F. H. Murphy, and E. A. Stohr. 1989. "A Graphics Interface for Linear Programming," Communications of the ACM 32:996-1012.
- MacGregor, J. M., and E.S. Lee. 1988. "A Feature Matching Approach to the Retrieval of Graphical Information." Behavior and Information Technology 7(4):457-465.
- Magnavox, Inc. 1985. Appendix B - Application of Data Processing Networking

Techniques for Army Command and Control Systems (ACCS) Task 1 Subtask 4.

Report No. 11-89 on Contract No. DAAK11-84-D-0003, Magnavox, Inc., Fort Wayne, Indiana.

Main, R. and D. Paulson. 1988. Guidelines for the Development of Military Training Decision Aids. NPRDC TR 88-16. Navy Personnel Research and Development Center, San Diego, California.

Mallary, T. C. 1985. "Design of the Human-Computer Interface for a Computer Aided Design Tool for the Normalization of Relations." Master's Thesis, Air Force Institute of Technology, Wright Air Force Base, Ohio.

Martin, J. 1983. Managing the Data-Base Environment. Prentice Hall, Englewood Cliffs, New Jersey.

Masiaszek, L. A. 1990. Database Design and Implementation. Prentice Hall, New York.

Matthews, M. L. 1987. "The Influence of Color on CRT Reading Performance and Subjective Comfort Under Operational Conditions." Applied Ergonomics 18.4:323-328.

McCann, C. 1988. Final Report from "Research Study Group 12 (RSG.12) on Computer-Human Interaction in Command and Control." Document AC/243 (Panel 8/RSG.12) D/7, DCIEM, Downsview, Ontario.

McCann, P. H. 1983. Methods for Improving the User-Computer Interface. Report No. NPRDC TR 83-29, Navy Personnel Research and Development Center, San Diego, California.

McKeown, P. E., J. J. Fallesen, M. S. Perkins, and C. G. Ross. 1991. Operations Planning Tools (OPT) Functional Description. U.S. Army Research Institute Product 91-09 (AD-A235 665), Fort Leavenworth, Kansas.

McNeese, M. D., and L. Katz. 1986. "Legibility Evaluation of a Large-Screen Display System Under Medium Ambient Illumination." In Society for Information Display 86 Digest, pp. 142-145. SID, New York.

- Milgram, P., D. Drascic, and J. J. Grodski. 1991. "Enhancement of 3-D Video Displays by Means of Superimposed Stereo-Graphics." In Proceedings of the Human Factors Society - 35th Annual Meeting, pp. 1457-1461. HFS, Santa Monica, California.
- Milheim, W.D., and B. L. Martin. 1991. "Theoretical Basis for the Use of Learner Control: Three Different Perspectives." Journal of Computer-Based Instruction 18(3):99-105.
- Miller, P. J. 1991. NMR Display Prototype Experiment Results. Technical Report No. AD-A239 131, Defense Technical Information Center, Alexandria, Virginia.
- Minasi, M. 1990. "Bayes and Simple Expert Systems." AI Expert, pp. 13-15.
- Mitchell, D. K., and K. P. Kysor. 1992. A Preliminary Evaluation of the Prototype Tactical Computerized Interactive Display. Technical Memorandum 2-92. U.S. Army Human Engineering Laboratory, Aberdeen Proving Ground, Maryland.
- Mittal, S. 1985. "Knowledge Acquisition from Multiple Experts." AI Magazine, pp. 32-36.
- Miyashita, T., and T. Uchida. 1990. "Cause of Fatigue and its Improvement in Stereoscopic Displays." Proceedings of Society for Information Display, 31(3):249-254. SID, New York.
- Moran, T. P. 1981. "The Command Language Grammar: A Representation for the User of Interactive Computer Systems." International Journal of Man-Machine Studies 15:3-50.
- Morse, R. S. 1985. "Glare Filter Preference: Influence of Subjective and Objective Indices of Glare, Sharpness, Brightness, Contrast and Color." In Proceeding of the Human Factors Society - 29th Annual Meeting, pp.782-786. HFS, Santa Monica, California.
- Muracka, T., M. Kawamura, and H. Uesako. 1989. "Readability on the Positive Type Liquid Crystal Display Devices with Multinumerals Influenced by the Irradiation Illuminances." In Work With Computers: Organizational, Management, Stress and Health Aspects, eds. M. J. Smith and G. Salvendy, pp. 542-548. Elsevier Science Publishers B.V., Amsterdam.

Muter, P. and C. Mayson. 1986. "The Role of Graphics in Item Selection from Menus." Behavior and Information Technology 5:89-95.

National Institute of Standards and Technology (NIST). 1991a. Application Portability Profile (APP) The U.S. Government's Open System Environment Profile OSE/1 Version 1.0 (FIPS Pub 151-1), National Institute of Standards and Technology, Gaithersburg, Maryland.

National Institute of Standards and Technology (NIST). 1991b. Government Open Systems Interconnection Profile (GOSIP), Version 1.0 (FIPS Pub 146-1), National Institute of Standards and Technology, Gaithersburg, Maryland.

National Institute of Standards and Technology (NIST). 1990a. POSIX, Portable Operating System Interface for Computer Environments (IEEE 1003.1-1988) (FIPS Pub 151-1), National Institute of Standards and Technology, Gaithersburg, Maryland.

National Institute of Standards and Technology (NIST). 1990b. "User Interface Component of Applications Portability Profile." Federal Information Processing Standard (FIPS) 158. NIST, Gaithersburg, Maryland.

Nes, F. 1986. "Space, Color, and Typography on Visual Display Terminals." Behavior and Information Technology 5(2):99-118.

Nickerson, R. S. 1986. Using Computers: The Human Factors of Information Systems. The MIT Press, Cambridge, Massachusetts.

Nicol, A. 1990. "Interface for Learning: What Do Good Teachers Know That We Don't?" In The Art of Human-Computer Interface Design, ed. B. Laurel, pp. 113-122. Addison-Wesley, Reading, Massachusetts.

Nielsen, J. 1987. "A User Interface Case Study of the Macintosh." In Cognitive Engineering in the Design of Human-Computer Interaction and Expert Systems. ed. G. Salvendy, pp. 241-248. Elsevier Science Publishers B.V., Amsterdam.

Norman, K. L., and J. P. Chin. 1988. "The Effect of Tree Structure on Search in a Hierarchical Menu Selection System." Behavior and Information Technology 7(1):51-65.

North Atlantic Treaty Organization (NATO). 1990. North Atlantic Treaty Organization Standardization Agreement 2019, Military Symbols for Land Based Systems. U.S. Navy, Washington, D.C.

Ogden, W. C., and S. R. Brooks. 1983. "Query Languages for the Casual User: Exploring the Middle Ground between Formal and Natural Languages." In Human Factors in Computing Systems: Proceedings of the CHI '83 Conference, ed. A. Janda. North Holland Publishing Company, Amsterdam.

O'Keefe, R. M. 1989. "The Evaluation of Decision-Aiding Systems: Guidelines and Methods." Information and Management 17:217-226.

Olsen, L. A., and T. N. Huckin. 1991. Technical Writing and Professional Communication (2nd ed.). McGraw-Hill, New York.

Olson, J. M. 1987. "Color and the Computer in Cartography." In Color and the Computer, ed. H. J. Durrett, pp. 205-219. Academic Press, Inc., San Diego, California.

O'Malley, C., P. Smolensky, L. Bannon, E. Conway, J. Graham, J. Sokolov, and M. L. Montry. 1983. "A Proposal for User-Centered System Documentation." In HMI Project. User-Centered System Design: Papers for the CHI '83 Conference on Human Factors in Computer Systems, p. 4-8. DTIC No. AD 136131. (Technical Report). Office of Naval Research/Personnel and Training, Arlington, Virginia.

Open Software Foundation (OSF). 1990. OSF/Motif Style Guide. Prentice Hall, Englewood Cliffs, New Jersey.

Osborn, P. B., and W. H. Zickefoose. 1990. "Building Expert Systems From the Ground Up." AI Expert, pp. 28-35.

OSF - see Open Software Foundation

Otte, F. H. 1982. "Consistent User Interface." In Human Factors and Interactive Computer Systems, pp. 261-275. Ablex, Norwood, New Jersey.

Owen, D. 1987. "Direct Manipulation and Procedural Reasoning." In Cognitive Engineering in the Design of Human-Computer Interaction and Expert Systems, ed. G. Salvendy, pp. 349-356. Elsevier Science Publishers, B.V., Amsterdam.

Paap, K. R., and R. J. Roske-Hofstrand. 1988. "Design of Menus." In Handbook of Human-Computer Interaction, ed. M. Helander, Elsevier Science Publishers, B.V., Amsterdam.

Paap, K. R., and R. J. Roske-Hofstrand. 1986. "The Optimal Number of Menu Options Per Panel." Human Factors 28(4):377-385.

Parker, S. P., ed. 1989. McGraw-Hill Dictionary of Scientific and Technical Terms. 4th ed. McGraw-Hill, New York.

Parkinson, S. R., M. D. Hill, N. Sisson, and C. Viera. 1988. "Effects of Breadth, Depth, and Number of Responses on Computer Menu Search Performance." International Journal of Man-Machine Studies 28:683-692.

Parrish, R. N., J. L. Gates, and S. J. Munger. 1981. Design Guidelines and Criteria for User/Operator Transactions with Battlefield Automated Systems Volume IV: Provisional Guidelines and Criteria. Technical Report 537, U.S. Army Research Institute for the Behavioral and Social Sciences, Alexandria, Virginia.

Parsaye, K., M. Chignell, K. Setrag, and H. Wong. 1990. "Intelligent Databases." AI Expert, Mar. 1990, pp. 38-47, Miller Freeman, Inc.

Patricia Seybold's Office Computing Group. 1989. "Patricia Seybold's Office Computing Report." April 1989. 12(4):1(9).

Pavard, B. 1987. "Design of Graphic Dialogue Without Syntactic Constraints." In Cognitive Engineering in the Design of Human-Computer Interaction and Expert Systems, ed. G. Salvendy, pp. 349-356. Elsevier Science Publishers B.V., Amsterdam.

Payne, S. J. 1983. "Readability of Liquid Crystal Displays: A Response Surface." Human Factors Society, Inc. 25(2):185-190. Santa Monica, California.

Petrun, C. J., W. Hennon, and M. MacDonald. 1985. "An Examination of the Relative Legibility of Text on One Line Vacuum Fluorescent and Liquid Crystal Displays." In Proceedings of the Human Factors Society - 29th Annual Meeting, pp.1122-1124. HFS, Santa Monica, California.

Pew, R. W. 1988. "Human Factors Issues in Expert Systems." In Handbook of Human-Computer Interaction, ed. M. Helander, pp. 931-940. Elsevier Science Publishers B.V., Amsterdam.

Pleshko, P. 1991. "AC Electroluminescent Display Technology: Challenges and Potential." Proceedings of Society for Information Display, 32(2):106-108. SID, New York.

Rancourt, J., W. Grenawalt, M. W. Hunter, and H. L. Snyder. 1986. "Quantitative Evaluation of the Effect of an Antireflection Filter." In Society for Information Display 86 Digest, pp. 420-423. SID, New York.

Raum, H. G. 1988. Design and Implementation of an Expert User Interface for the Computer Aided Prototyping System. Thesis for U.S. Naval Postgraduate School, Monterey, California.

Raymond, E. S., ed. 1991. The New Hacker's Dictionary. The MIT Press, Cambridge, Massachusetts.

Reedy, A.E., D.L. Smith, and W.G. Bail. 1992. Software Architecture Framework (Draft), Defense Systems Information Agency/Center for Information Management, McLean, Virginia.

Reger, J. J., H. L. Snyder, W. W. Farley. 1989. "Legibility of Emissive and Non-Emissive Displays Under Florescent and Daylight Illumination." In Society for Information Display 89 Digest, pp. 364-367. SID, New York.

Reinhart, W. F. 1990. "Effects of Depth Cues on Depth Judgments Using a Field-Sequential Stereoscopic CRT Display." Unpublished Doctoral Dissertation, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

Relles, N. 1981. "A User Interface for On-line Assistance." In Proceedings of the Fifth International Conference on Software Engineering, pp. 400-408. Institute for Electrical and Electronics Engineers (IEEE), New York.

Relles, N., and L. A. Price. 1981. "A User Interface for On-line Assistance." In Proceedings of the Fifth International Conference on Software Engineering. Institute for Electrical and Electronics Engineers (IEEE), New York.

Riedel, S. L. 1988. "User Acceptance and Field Implementation of Decision Support Systems." Research Report 1477. U.S. Army Research Institute Field Office, Fort Leavenworth, Kansas.

Ripley, G. D. 1989. "DVI--A Distal Multimedia Technology". Communications of the ACM 32:811-822.

Roach, J., R. Hartson, R. W. Ehrich, T. Yuntan, and D. H. Johnson. 1982. "DMS: A Comprehensive System for Managing Human-Computer Dialogue." In Human Factors in Computer Systems: Proceedings March 15-17, 1982, Gaithersburg, Maryland.

Rockley, A. 1987. "Online Documentation: From Proposal to Finished Product." In Proceedings of 34th International Technical Communication Conference, ATA 58-61. Society for Technical Communication, Washington, D.C.

Roth, J.L., J. A. Fitzpatrick, R. E. Warm, and J. L. Ditzian. 1988. Implementing Embedded Training (ET): Volume 5 of 10: Designing the ET Component. ARI Research Product No. 88-28, NTIS No. AD-A205 697. U.S. Army Research Institute for the Behavioral and Social Sciences, Alexandria, Virginia.

Rupp, B. A. 1981. "Visual Display Standards: A Review of Issues." Proceedings of Society for Information Display, 22(1):63-72. SID, New York.

Salvendy, G. 1987. Cognitive Engineering in the Design of Human-Computer Interaction and Expert Systems. Elsevier Science Publishers B.V., Amsterdam.

Schauer, U. 1983. "The Integrated Data Analysis and Management System - A Generator for Enduser Systems." Enduser Systems and Their Human Factors eds. A. Blaser and M. Zoeppriz, pp. 30-61. Springer-Verlag, Berlin.

Schmitz, J. D., G. D. Armstrong, and J. D. Little. 1990. "Cover Story - Automated News Finding in Marketing." Interfaces 20(6):29-38.

Schur, S. 1988. "The Intelligent Database." AI Expert, Jan. 1988, pp. 26-34, Miller Freeman, Inc.

Schwartz, J. P. 1983. "Lack of Guidance for Decision Aid Interface Design." SIGCHI Bulletin 15:13-17.

Sellen, A., and A. Nicol. 1990. "Building User-Centered On-line Help." In The Art of Human-Computer Interface Design, ed. B. Laurel, pp. 143-153. Addison-Wesley, Reading, Massachusetts.

Shaw, B., and M. McCauley. 1985. Personal Computer Dialogue: A Human Engineering Data Base Supplement. Technical Report No. AFAMRL-TR-85-013. U.S.A.F. Aerospace Medical Research Laboratory, San Antonio, Texas.

Shields, S. E., and W. P. Bleha. 1991. "Liquid Crystal Light Valves for Projection

Displays." In Society of Photoptic Instrumentation Engineers, 1455:225-236. SPIE Press, Bellingham, Washington.

Shneiderman, B. 1991. "Visual User Interfaces for Information Exploration", made available at the TAE Ninth User's Conference, New Carrollton, Maryland. CAR-TR-577, CS-TR-2748, Human-Computer Interaction Laboratory and Department of Computer Sciences, University of Maryland, College Park, Maryland.

Shneiderman, B. 1988. "We Can Design Better User Interfaces: A Review of Human-Computer Interaction Styles." Ergonomics 31(5):699-710.

Shneiderman, B. 1987. Designing the User Interface: Strategies for Effective Human-Computer Interaction. Addison-Wesley, Reading, Massachusetts.

Shneiderman, B. 1984. "The Future of Interactive Systems and the Emergence of Direct Manipulation." In Human Factors and Interactive Computer Systems, ed. Y. Vassiliou, pp. 1-27. Ablex Publishing Corporation, Norwood, New Jersey.

Shneiderman, B. 1982. "Designing Computer System Messages." Communications of the ACM, 25(9):604-605.

Shneiderman, B. 1978. "Improving the Human Factors Aspect of Database Interactions." ACM Transactions on Database Systems 3(4):417-439.

Shurtleff, D. A., W. F. Wuersch, and J. G. Rogers. 1982. "Applications of Large-Screen Display Legibility Criteria." In Society for Information Display 82 Digest, pp. 202-203. SID, New York.

Shurtleff, D. A., W. F. Wuersch, and J. G. Rogers. 1981. "How to Make Large Screen Displays Legible." In Proceedings of the Human Factors Society - 25th Annual Meeting, pp. 149-153. HFS, Santa Monica, California.

Sidorsky, R. 1984. Design Guidelines for User Transactions with Battlefield Automated Systems: Prototype for a Handbook. DTIC AD-A153 231, U.S. Army Research Institute for the Behavioral and Social Sciences, Alexandria, Virginia.

Silverstein, L. D., R. W. Monty, J. W. Huff, and K. L. Frost. 1987. Image Quality and Visual Simulation of Color Matrix Displays. Technical Paper No. 871789, The Engineering Society for Advancing Mobility Land, Sea, Air, and Space, Long Beach, California.

Sisson, N., S. R. Parkinson, and K. Snowberry. 1986. "Consideration of Menu Structure and Communication Rate for the Design of Computer Menu Displays." International Journal of Man-Machine Studies 25:479-489.

Slominski, S.E., and I. R. Young. 1988. A User Friendly Design of an Interactive Prototype for the Maintenance and Monitoring of Civilian Training Records. Unpublished Master's Thesis, Naval Postgraduate School, Monterey, California.

Smith, S. L., and J. N. Mosier. 1984. A Design Evaluation Checklist for User Computer Interface Software. MITRE Corporation, Bedford, Massachusetts.

Smith, S. L., and J. N. Mosier. 1986. Guidelines for Designing User Interface Software. MITRE Corporation, Bedford, Massachusetts.

Snyder, H. L. 1980. "Human Visual Performance and Flat Panel Display Image Quality." HFL-80-1/ONR-80-1, Human Factors Laboratory, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

Snyder, H. L. 1984. "Ergonomics Database for Visual Displays and VDUs." In Ergonomic Data for Equipment Design, ed. H. Schmidtke, pp. 219-234. Plenum Press, New York.

Snyder, H. L. 1987. "The ANSI Human Factors Standard for Visual Display Terminal Workstation: A Process and Progress Report (Invited Address)." In Society Information Display 87 Digest, pp. 14-17. SID, New York.

Snyder, H. L. 1988. "Image Quality." In Handbook of Human-Computer Interaction, ed. M. Helander, pp. 437-474. Elsevier Science Publishers B.V., Amsterdam.

Sormunen, E. 1987. "A Knowledge-Based Intermediary System for Information Retrieval," in Knowledge Engineering Expert Systems and Information Retrieval, ed. I. Wormell, pp. 59-73. Taylor Graham, London.

Steele, C. A., J. Harrold, J. P. Stanton, and R. Daley. 1987. "Performance of Laser-Addressed Liquid Crystal Map Overlay Display." In Society of Photoptic Instrumentation Engineers, 760:70-73. SPIE Press, Bellingham, Washington

Strategic Air Command (SAC). 1990. Intelligence Data Handling System (IDHS) Man-Machine Interface (MMI) Style Guide, Version 1.0. Washington, D.C.

Sun Microsystems, Inc. 1990. The Open Look Graphical User Interface Application

Style Guidelines. AT&T, New Jersey.

Suntola, T. 1989. "VLSI and Computer Peripherals: Thin Film EL-Display." Proceedings of the Institution of Electrical and Electronics Engineers, pp.2-32/2-35. IEEE Computer Society Press, Washington, D.C.

Tannas, L. E., ed. 1985. Flat-Panel Displays and CRTs. Van Nostrand Reinhold, New York.

Tannenbaum, A. 1990. "Installing AI Tools Into Corporate Environments." AI Expert, pp. 54-59.

Tenopir, C., and G. Lundeen. 1988. Managing Your Information. Neal-Schuman Publishers, New York.

Thierauf, R. J. 1988. User-Oriented Decision Support Systems. Prentice Hall, Englewood Cliffs, New Jersey.

Thomas, J. C. 1983. "Psychological Issues in Data Base Management." In Designing for Human-Computer Communications, eds. M. E. Sime and M. J. Coombs, pp. 169-184. Academic Press, London.

Thomas, J.C., and J. M. Carroll. 1981. "Human Factors in Communication." IBM Systems Journal 20(2):236-263.

Thorell, L. G., and W. J. Smith. 1990. Using Computer Color Effectively: An Illustrated Reference. Prentice Hall, Englewood Cliffs, New Jersey.

TRW. 1990a. The AWIS Common User/System Interface Style Guide. AWIS Program Documentation.

TRW. 1990b. UTACCS, Soldier-Machine Development Standards. UTACCS Program Documentation.

Tullis, T. S. 1988. "Screen Design." In Handbook of Human-Computer Interaction, ed. M. Helander, pp. 377-411. Elsevier Science Publishers B.V., Amsterdam.

U.S. Department of the Army. 1987. Map Reading and Land Navigation. FM 21-26, Washington, DC.

U.S. Department of the Army. 1985a. Authorized Abbreviations and Brevity Codes. AR310-50, Army UPDATE Publications, Washington, D.C.

U.S. Department of the Army. 1985b. Operational Terms and Symbols. FM 101-5-1, U.S. Army Combined Arms Center, Fort Leavenworth, Kansas.

U.S. Department of the Army. 1984. Human Engineering Guide to Equipment Design. FM 21-26 (sec 11), eds. H. P. Van Cott and R. G. Kinkade. John Wiley and Sons, New York.

U.S. Department of Defense (DoD). 1992a. Department of Defense Human Computer Interface Style Guide (Version 1.0), Defense Systems Information Agency/Center for Information Management, McLean, Virginia.

U.S. Department of Defense (DoD). 1992b. Standards-Based Architecture Planning Handbook, Draft Version 1.0. Produced by DMR Group, Inc. for DISA XI, Washington, D.C.

U.S. Department of Defense (DoD). 1992c. Technical Reference Model for Information Management, Version 1.2, Defense Information Systems Agency/Center for Information Management, McLean, Virginia.

U.S. Department of Defense (DoD). 1991a. Department of Defense Intelligence Information Systems Style Guide, Department of Defense Intelligence Information Systems (DODIIS) Management Board, Arlington, Virginia.

U.S. Department of Defense (DoD). 1991b. DoD Directive 5000.1 Defense Acquisition, U.S. Department of Defense, Washington, D.C.

U.S. Department of Defense (DoD). 1991c. DoD Instruction 5000.2. Defense Acquisition Management Policy and Procedures, U.S. Department of Defense, Washington, D.C.

U.S. Department of Defense (DoD). 1990. Military Training Programs. MIL-STD 1379D, Naval Sea Systems Command, SEA 5523.

U.S. Department of Defense (DoD). 1989a. Department of Defense Dictionary of Military and Associated Terms. DoD, Washington, D.C.

U.S. Department of Defense (DoD). 1989b. Human Engineering Design Criteria for Military Systems, Equipment, and Facilities. MIL-STD-1472D, U.S. Army Missile Command, Huntsville, Alabama.

U.S. Department of Defense (DoD). 1989c. Human Engineering Guidelines for Management Information Systems. DOD-HDBK-761A, DoD, Washington, D.C.

U.S. Department of Defense (DoD). 1988. DoD Directive 7920.1. Life-Cycle Management of Automated Information Systems (AISs), U.S. Department of Defense, Washington, D.C.

Ullman, J. D. 1988. Principles of Database and Knowledge-Base Systems. Computer Science Press, Rockville, Maryland.

UNIX System Laboratories, Inc. 1991. Open Look Graphical User Interface User's Guide. Prentice Hall, Inc., Englewood Cliffs, New Jersey.

Urban, C. D. 1990. "Design and Evaluation of a Tactical Decision Aid." Proceedings IEEE International Conference on Systems, Man, and Cybernetics. DTIC No. AD-A232 001, IEEE, New York.

Vance, D. W. 1987. "Image Distortions in Video Projection." In Large Screen Projection Displays, 760:54-59. SPIE Press, Bellingham, Washington.

Van Cott, H. P., and R. G. Kinkade, eds. 1984. Human Engineering Guide to Equipment Design. John Wiley and Sons, New York.

Vassiliou, Y., ed. 1984. Human Factors and Interactive Computer Systems. Ablex Publishing Corporation, Norwood, New Jersey.

Vasta, J. A. 1985. Understanding Data Base Management Systems. Wadsworth Publishing Company, Belmont, California.

Veith, R. H. 1988. Visual Information Systems: The Power of Graphics and Video. G.K. Hall & Co., Boston.

Veron, H. 1988. A Resolution Measurement Technique for Large Screen Displays. Technical Report No. M88-59, The MITRE Corp. Bedford, Massachusetts.

- Veron, H., D. A. Southard, J. R. Leger, and J. L. Conway. 1990. 3D Displays for Battle Management. Technical Report No. AD-A223 142, Defense Technical Information Center, Alexandria, Virginia.
- Verplank, W. L. 1988. "Graphic Challenges in Designing Object-Oriented User Interfaces." In Handbook of Human-Computer Interaction, ed. M. Helander, pp. 365-375. Elsevier Science Publishers B.V., Amsterdam.
- Waern, Y., and C. Rollenhagen. 1983. "Reading Text From Visual Display Units (VDUs)." In International Journal of Man-Machine Studies 18, Academic Press, London.
- Walker, J. H. 1987. "Issues and Strategies for Online Documentation." IEEE Transactions on Professional Communication, PC-30 (4):235-248.
- Walrath, J. D. 1989. "Aiding the Decision-Maker: Perceptual and Cognitive Issues at the Human-Machine Interface." Technical Note 15-89. U.S. Army Human Engineering Laboratory (DTIC No. AD-A217 862), Aberdeen Proving Ground, Maryland.
- Wehrer, W., and E. E. Mitchamore. 1992. "Technology/Marketing Case Study: Plug-and-play IR Touch Screens." Information Display, pp. 10-12. SID, New York.
- Weingaertner, S. T., and A. H. Levis. 1988. "Evaluation of Decision Aiding in Submarine Emergency Decision-Making." IFAC Conference on the Analysis, Design and Evaluation of Man-Machine Systems, Oulu, Finland, pp. 195-201.
- Wenger, E. 1987. Artificial Intelligence and Tutoring Systems. Morgan Kaufman Publishers, Los Altos, California.
- Wexelblat, R. L. 1989. "On Interface Requirements for Expert Systems." AI Magazine 10:66-78.
- Williams, K.E., C. J. Hamel, and L. B. Shrestha. 1987a. An Evaluation of Characteristics Contributing Towards Ease of User-Computer Interface in a Computer-Aided Instruction Exercise. Technical Report TR87-030, Naval Training Systems Center, Orlando, Florida.

Williams, K.E., C. J. Hamel, and L. B. Shrestha. 1987b. CAI Evaluation Handbook: Guidelines for the User Interface Design for Computer-Aided Instruction. Technical Report No. TR87-033, Naval Training Systems Center, Orlando, Florida.

Williams, R. D. 1990. "Volume 3D Display Technology." In Stereographics, 17th International Conference on Computer Graphics and Interactive Techniques, Association for Computing Machinery, Special Interest Group on Graphics, Dallas Texas.

Wilson, M., P. Barnard, and A. MacLean. 1990. An Investigation of the Learning of a Computer System. In Cognitive Ergonomics: Understanding, Learning and Designing Human-Computer Interaction, ed. P. Falzon, pp. 151-172. Academic Press, London.

Woodson, W. E., B. Tillman, and P. Tillman. 1992. Human Factors Design Handbook: Information and Guidelines for the Design of Systems, Facilities, Equipment, and Products for Human Use, Second Edition. McGraw-Hill, Inc., New York.

Wormell, I., ed. 1987. Knowledge Engineering Expert Systems and Information Retrieval. Taylor Graham, London.

Wyatt, A. L. 1990. Computer Professional's Dictionary. Osborne McGraw-Hill, Berkeley, California.

Yamazaki, T., K. Kamijo, and S. Fukuzumi. 1990. "Quantitative Evaluation of Visual Fatigue Encountered in Viewing Stereoscopic 3D Displays: Near-Point Distance and Visual Evoked Potential Study." Proceedings of Society for Information Displays, 31(3):245-247. SID, New York.

Yeh, Y., and L. D. Silverstein. 1991. "Human Factors for Stereoscopic Color Displays." In Society for Information Display 91 Digest, pp. 826-829. SID, New York.

Yeh, Y., and L. D. Silverstein. 1989. "Using Electronics Stereoscopic Color Displays: Limits of Vision and Depth Discrimination." Three-Dimensional Visualization and Display Technologies, Proceedings of SPIE International Society for Optical Engineering, Vol. 1083, pp. 196-204. SPIE Press, Bellingham, Washington.

Zachary, W. W. 1988. "Decision Support Systems: Designing to Extend the Cognitive Limits." In Handbook of Human-Computer Interaction, ed. M. Helander, pp. 997-1030. Elsevier Science Publishers B.V., Amsterdam.

Ziegler, J. E., and K. P. Fähnrich. 1988. "Direct Manipulation." In Handbook of Human-Computer Interaction, ed. M. Helander, pp. 123-134. Elsevier Science Publishers B.V., Amsterdam.

Zmud, R. W. 1978. "Concepts, Theories and Techniques: An Empirical Investigation of the Dimension of the Concept of Information." Decision Sciences 9:187-195.

Addendum 1

User Interface Specifications for Navy Command and Control Systems

**USER INTERFACE SPECIFICATIONS
FOR NAVY COMMAND AND CONTROL SYSTEMS**

**Version 1.1
June 1992**

**Kathleen Fernandes, Ph.D.
Naval Command, Control, and Ocean Surveillance Center
Research, Development, Test, and Evaluation Division**

FOREWORD

The purpose of the specifications contained herein is to define a common "look and feel" for the user interface of Navy command and control (C²) systems. Version 1.1 of the specifications provides the following major enhancements over the previous version of the document:

1. In general, when the specifications were modified, the changes were made to increase consistency with standard Motif appearance and behavior. Where necessary, deviations were made to ensure compliance with IEEE recommendations on graphical user interface drivability. A list of changes is included with this version. As with the initial document, topics not yet covered in version 1.1 are indicated in footnotes and will be addressed in future versions of the specifications.

2. The graphics have been upgraded to be more consistent with the "look and feel" defined by the specifications and have been integrated into the body of the document.

3. Motif resource settings that implement the "look and feel" called for by the specifications are included in this version of the document. These settings define the general appearance and behavior of both system- and application-level windows and the objects that are displayed in them.

4. A checklist of the mandatory and recommended provisions contained in the specifications is provided as an appendix to the document.

5. An index of topics addressed in the specifications is included in this version.

The specifications have been submitted to the Navy Command and Control Systems Ashore Program (PMW-161), Space and Naval Warfare Systems Command for approval and adoption by the Navy. Until formal adoption occurs, application developers should contact the developing agency with responsibility for the system(s) in which the application will be integrated to determine if compliance with this document is required. In addition, the document will be circulated to Navy fleet commands to solicit input on the acceptability of the specifications from an operational perspective. When the specifications are completed and adopted by the Navy, they will be published and distributed as an NRaD technical document.

Comments concerning the specifications should be directed to Dr. Kathleen Fernandes, NCCOSC, RDT&E Division, Code 423, San Diego, CA 92152, (619) 553-9224 (same for DSN), fax: (619) 553-5136, e-mail: fernande@nosc.mil.

Summary of changes by section.

<u>Section</u>	<u>Change</u>
General	Change "text cursor" to "text insertion cursor."
General	Where appropriate, quote rather than paraphrase the Motif Style Guide; include relevant version of Style Guide being referenced.
General	Fix typographical and grammatical errors.
1.3.1 & footnote	Include more detail on hardware and software assumptions.
1.3.3, 1st para.	Clarify Motif "look and feel" of COTS software included in systems.
1.4 & footnote	Include compliance of specifications with DoD style guide and IEEE user interface recommendations.
1.6, 1st para.	Address compliance with specifications if using COTS tools or software.
1.7	Include reference to checklist provided in Appendix B.
1.8	Update references.
2.1 footnote & 2nd para.	Clarify discussion on interchangeability between keyboard and pointing device.
2.1.1.3, 2nd para.	Increase flexibility in assigning a function to Middle button on the pointing device.
2.1.2.3, .4, & .5	Add sections on navigation, modifier, and function key assignments.
2.2.2	Revise object selection/deselection to comply with IEEE user interface recommendations.
2.2.3.1, 1st para.	Delete reference to separate parts of double-click action.
2.2.4, 2nd para.	Add capability to view the content of the clipboard buffer.
2.2.4.1	Add paragraph listing drag-and-drop rules from the Motif Style Guide.
2.2.4.2	Correct the definition of the Clear command.
2.4.2, 2nd para.	Add visual feedback requirement for objects such as menu items.
2.4.3	Clarify feedback provided to users during lengthy operations; incorporate IEEE recommendations on system response time.
2.4.5, 1st para.	Modify to require confirmation of all destructive (vice potentially destructive) actions.
3.1 & 3.3	Modify to include full range of window management functions in the Window Menu.

Summary of changes by section (continued).

<u>Section</u>	<u>Change</u>
3.2.2, 2nd para.	Clarify processing that occurs in an active window without input focus.
3.2.3	Delete reference to icon box.
3.3.1	Delete reference to moving windows between screens.
3.3.1 - 3.3.6	Modify to include window manipulation using Window Menu functions; delete requirement to use icon box in section 3.3.4.
Table 4-1	Expand list of keyboard accelerators to include those defined in the Motif Style Guide.
4.1.4, 2nd para.	Modify section on the design of cascading menus.
4.2, 3rd para.	Modify wording describing availability of pop-up menus.
5.1, 2nd para.	Clarify wording on normal push button behavior.
Table 5-1	Change "temporary buffer" to "clipboard buffer."
5.1.2	Clarify wording on appearance of default push button.
5.1.3	Add section on modifying normal push button behavior.
5.5.3	Clarify requirement to provide speed and incremental search capabilities.
6.1	Clarify requirement to display a message when system is unavailable.
6.2.1, 2nd para.	Delete reference to icon box.
6.2.3	Include color names and RGB values for classification markings.
6.3	Change section title from "System Menu Bar" to "System Menu."
6.3.4, 2nd para.	Revise wording on appearance of menu option when only one version of an application can be running.
6.4.1, 2nd & 3rd para.	Clarify file manipulation capabilities that should be provided by a system; add capability to temporarily suspend a session.
7.1.1.3.3	Delete reference to turning on/off action icons.
7.1.2.2	Modify to require a warning message window to confirm all destructive (vice potentially destructive) actions.
7.2.1 & Table 7-1	Revise color scheme to be consistent with Motif resource settings in Appendix A.
7.3.1, 2nd para.	Add recommendations from Kobara on text font style.

Summary of changes by section (continued).

<u>Section</u>	<u>Change</u>
7.5.5, 2nd & 3rd para.	Require (vice recommend) an auditory signal when alerting users to critical conditions; add IEEE recommendation for flash coding.
8.6.1, 4th para.	Add capability to select among multiple symbol sets in a map window.
8.7	Change section title from "Message Communications Windows" to "Message Handling Windows."
Appendix A	Add Motif resource settings for system- and application-level windows and objects.
Appendix B	Add checklist of mandatory and recommended provisions contained in the specifications.
Index	Add subject index to topics addressed in the specifications.
Graphics	Revise to be more consistent with provisions of the specifications.

CONTENTS

1.0	INTRODUCTION	1-1
1.1	BACKGROUND	1-1
1.2	PURPOSE	1-1
1.3	ASSUMPTIONS	1-2
1.3.1	Hardware and Software Configuration	1-2
1.3.2	System Functionality	1-2
1.3.3	System Integration	1-3
1.3.4	System Users	1-4
1.4	SCOPE	1-4
1.5	INTENDED AUDIENCE	1-6
1.6	COMPLIANCE	1-6
1.7	DOCUMENT OVERVIEW	1-7
1.8	REFERENCES	1-7
2.0	USER-COMPUTER INTERACTION	2-1
2.1	INPUT DEVICES	2-1
2.1.1	Pointing Device Input	2-1
2.1.1.1	The Pointer	2-1
2.1.1.2	Pointer Shapes	2-2
2.1.1.3	Pointing Device Buttons	2-4
2.1.2	Keyboard Input	2-4
2.1.2.1	Keyboard Functions	2-4
2.1.2.2	The Text Insertion Cursor	2-4
2.1.2.3	Navigation Key Assignments	2-6
2.1.2.4	Modifier Key Assignments	2-7
2.1.2.5	Function Key Assignments	2-8
2.1.3	Alternative Input Devices	2-8
2.2	OBJECT SELECTION AND MANIPULATION	2-8
2.2.1	Object-Action Selection	2-8
2.2.2	Selecting Objects	2-9
2.2.3	Other Types of Selection	2-9
2.2.4	Manipulating Objects	2-9
2.2.4.1	Dragging Objects to a New Location	2-11
2.2.4.2	Using Editing Commands	2-12
2.3	TEXT ENTRY.....	2-14
2.4	INTERACTIVE CONTROL	2-14
2.4.1	User Control of Interaction	2-14
2.4.2	Immediate Feedback	2-14
2.4.3	Response Time	2-15
2.4.4	Error Detection	2-16
2.4.5	Explicit Destruction	2-16
2.4.6	General "Undo" Capability	2-17

2.5	CONSISTENCY IN PERFORMING OPERATIONS	2-17
2.5.1	Consistency at the Application Level	2-17
2.5.2	Consistency at the System Level	2-18
3.0	WINDOWS	3-1
3.1	WINDOW COMPONENTS	3-1
3.2	WINDOW STATES	3-3
3.2.1	Input Focus	3-3
3.2.2	Open, Active, and Closed Windows	3-3
3.2.3	Iconified Windows	3-3
3.3	WINDOW MANIPULATION	3-4
3.3.1	Placing a Window	3-4
3.3.2	Moving a Window	3-4
3.3.3	Resizing a Window	3-4
3.3.4	Iconifying a Window	3-5
3.3.5	Lowering a Window.....	3-5
3.3.6	Closing a Window.....	3-6
3.4	TYPES OF WINDOWS	3-6
3.4.1	Primary Windows	3-6
3.4.2	Secondary Windows	3-8
3.4.2.1	Dialog Windows	3-8
3.4.2.2	Message Windows	3-8
4.0	MENUS	4-1
4.1	PULL-DOWN MENU	4-1
4.1.1	Menu Title	4-1
4.1.2	Menu Options	4-1
4.1.2.1	Types of Options	4-1
4.1.2.2	Wording of Options	4-3
4.1.2.3	Organization and Grouping	4-3
4.1.2.4	Available and Unavailable Options	4-4
4.1.3	Keyboard Accelerators	4-4
4.1.4	Cascading Menus	4-5
4.1.5	Menu Selection Methods	4-7
4.2	POP-UP MENU	4-8
4.3	OPTION MENU	4-8
5.0	CONTROLS	5-1
5.1	PUSH BUTTONS	5-1
5.1.1	Standard Vocabulary	5-2
5.1.2	Default Push Button Actions	5-2
5.1.3	Modifying Normal Push Button Behavior	5-8
5.2	RADIO BUTTONS	5-8
5.3	CHECK BUTTONS	5-9

5.4	TEXT ENTRY FIELDS	5-9
5.4.1	Format of Text Entry Fields	5-10
5.4.2	Distinguishing Text Entry Areas from Noneditable Text	5-11
5.4.3	Text Entry in a Text Entry Field	5-11
5.4.4	Moving between Text Entry Fields	5-12
5.5	LIST BOXES	5-12
5.5.1	Ordering Items in a List Box	5-13
5.5.2	Adding Items to a List Box	5-13
5.5.3	Searching Items in a List Box	5-14
5.5.4	Selecting Items in a List Box	5-14
5.5.5	Sorting Items in a Multi-Column List Box	5-16
5.6	SCROLL BARS	5-16
5.7	SCALES	5-19
6.0	SYSTEM-LEVEL WINDOWS	6-1
6.1	SYSTEM LOGIN	6-1
6.1.1	Login Procedure	6-1
6.1.2	The Login Window	6-1
6.1.3	Feedback during System Start-Up	6-2
6.2	THE SYSTEM WINDOW	6-2
6.2.1	Window Design	6-2
6.2.2	Window Appearance and Behavior	6-2
6.2.3	Classification Markings	6-5
6.3	THE SYSTEM MENU	6-6
6.3.1	Integration of Applications into System-Level Menus	6-6
6.3.2	Organization of the System Menu Bar	6-6
6.3.3	Consistency in Menu Titles between Systems	6-6
6.3.4	Access to Menu Options within an Application	6-7
6.3.5	Quick Access to System Functions	6-7
6.4	SYSTEM SUPPORT	6-8
6.4.1	System Support Functions	6-8
6.4.2	User-Specified Interface Settings	6-9
6.5	SYSTEM LOGOUT	6-9
6.5.1	User-Initiated Logout	6-9
6.5.2	Automatic Logout	6-10
7.0	APPLICATION-LEVEL WINDOWS	7-1
7.1	WINDOW ORGANIZATION	7-1
7.1.1	Primary and Dialog Windows	7-1
7.1.1.1	Window Title	7-1
7.1.1.2	Window Menu Bar	7-2
7.1.1.3	Main Area	7-2
7.1.1.3.1	Arrangement of Window Controls	7-2
7.1.1.3.2	Arrangement of Push Buttons	7-3

7.1.1.3.3 Arrangement of Action Icons	7-4
7.1.1.4 Message Bar	7-5
7.1.2 Message Windows	7-5
7.1.2.1 Progress Message Windows	7-6
7.1.2.2 Warning Message Windows	7-7
7.1.2.3 Error Message Windows	7-7
7.1.2.4 Information Message Windows	7-8
7.1.2.5 Question Message Windows	7-9
7.2 APPEARANCE AND BEHAVIOR OF OBJECTS	7-9
7.2.1 Color of Objects	7-9
7.2.2 Size of Objects	7-12
7.2.3 Consistent Appearance and Behavior	7-13
7.3 TEXT IN WINDOWS	7-13
7.3.1 Text Font, Size, and Readability	7-13
7.3.2 Capitalization	7-14
7.3.3 Acronyms and Abbreviations	7-14
7.3.4 Noneditable Text	7-15
7.3.5 Editable Text	7-15
7.4 COLOR USE IN WINDOWS	7-16
7.5 CONSIDERATIONS IN WINDOW DESIGN	7-17
7.5.1 Selecting Objects to Match User Actions	7-17
7.5.2 Arranging Information to Match User Actions	7-17
7.5.3 Arranging Information by Importance	7-19
7.5.4 Designing Windows to Minimize User Memory Load	7-19
7.5.5 Coding Critical Information in Windows	7-20
7.5.6 Dynamic Information in Windows	7-21
7.5.7 Consistency in Design Across Windows	7-21
7.6 WINDOW MANAGEMENT CONSIDERATIONS	7-22
7.6.1 Initial Window Appearance	7-22
7.6.2 Initial Window Size and Placement	7-22
7.6.3 Resizing	7-23
7.6.4 Order of Window Interaction	7-23
7.6.5 Processing in Iconified Windows	7-23
8.0 TASK-SPECIFIC WINDOWS	8-1
8.1 HELP WINDOWS	8-1
8.1.1 System-Level and Application-Level Help	8-1
8.1.2 Window Design	8-1
8.1.3 Window Content	8-2
8.1.4 Online Training	8-4
8.2 DATA ENTRY WINDOWS	8-4
8.2.1 Window Design	8-4
8.2.2 Data Field Format	8-6
8.2.3 Data Entry and Manipulation	8-6

8.2.4 Data Query	8-7
8.3 TEXT WINDOWS	8-7
8.3.1 Window Design	8-7
8.3.2 Text Manipulation	8-8
8.4 TABULAR DATA WINDOWS	8-8
8.4.1 Window Design	8-8
8.4.2 Window Content	8-9
8.5 GRAPHIC DISPLAY WINDOWS	8-11
8.5.1 Window Design	8-11
8.5.1.1 Line Graphs	8-11
8.5.1.2 Bar Charts and Histograms	8-13
8.5.1.3 Flow Charts	8-13
8.5.2 Manipulation of Graphical Data	8-13
8.6 MAP WINDOWS	8-16
8.6.1 Window Design	8-16
8.6.2 Map Manipulation	8-17
8.6.3 Automatic Updating	8-18
8.7 MESSAGE HANDLING WINDOWS	8-18
8.7.1 Message Preparation	8-18
8.7.2 Message Transmission	8-19
8.7.3 Message Receipt	8-20
APPENDIX A. MOTIF RESOURCE SETTINGS	A-1
APPENDIX B. USER INTERFACE SPECIFICATIONS CHECKLIST	B-1
INDEX	I-1

LIST OF TABLES

2-1 Pointer shapes	2-3
2-2 Text and graphics object selection and deselection	2-10
4-1 Standard keyboard accelerators	4-6
5-1 Action vocabulary	5-4
6-1 Appearance (color and 3-D effect) of the system window	6-3
7-1 Appearance (color and 3-D effect) of objects in application windows	7-10
A-1 Resources and settings for the Motif window manager	A-3
A-2 Resources and settings for Motif widgets	A-5

LIST OF FIGURES

1-1	National Institute of Standards and Technology user interface system reference model	1-5
2-1	I-beam pointer in a text entry area	2-5
2-2	Example pull-down menu containing edit commands	2-13
3-1	Standard components of a window	3-1
3-2	Example icon	3-5
3-3	Example primary and secondary windows	3-7
4-1	Example pull-down menu	4-1
4-2	Examples of menu option types	4-2
4-3	Example of logical ordering of menu options	4-3
4-4	Example of an unavailable menu option	4-4
4-5	Example keyboard accelerators	4-5
4-6	Example cascading menu	4-6
4-7	Organization of options in cascading menus	4-7
4-8	Example option menu	4-9
5-1	Example push buttons and action icons	5-1
5-2	Access manager window	5-3
5-3	Example default push button	5-7
5-4	Example group of radio buttons	5-8
5-5	Example group of check buttons	5-9
5-6	Example text entry field	5-10
5-7	Examples of text entry field delimiters	5-11
5-8	Example of automatic justification during text entry	5-12
5-9	Example list box	5-13
5-10	Example list and text entry field used in an incremental search	5-15
5-11	Example of a case-sensitive search	5-15
5-12	Example selection dialog box	5-16
5-13	Example list matrix	5-17
5-14	Components of a scroll bar	5-18
5-15	Examples of slider position within scroll regions	5-18
5-16	Example scale	5-19
6-1	Example login window	6-1
6-2	Example system window	6-3
6-3	Example colors for the system window	6-4
6-4	Maximum size of an application window	6-5
6-5	Example graphic navigation aid	6-8
7-1	Recommended format for primary and dialog windows	7-1
7-2	Example of grouping objects into control panels	7-3
7-3	Recommended format for action icons in windows	7-4
7-4	Example message bar in a dialog window	7-5
7-5	Example progress message window	7-6
7-6	Example warning message window	7-7

7-7	Example error message window	7-8
7-8	Example information message window	7-8
7-9	Example question message window	7-9
7-10	Example colors for interface objects	7-12
7-11	Example of breaking information into smaller text groups	7-15
7-12	Example of wording style	7-16
7-13	Map countries window	7-20
8-1	Example help window.....	8-2
8-2	Example of inconcise wording of help information	8-3
8-3	Example of concise wording of help information	8-3
8-4	Example data entry window	8-4
8-5	Consistency in layout of hardcopy form and data entry window	8-5
8-6	Example tabular display window	8-9
8-7	Example of data justification within columns	8-10
8-8	Example line graphs	8-12
8-9	Example bar charts	8-14
8-10	Example of labeling decision points in a flow chart	8-15
8-11	Example map window with identifying information	8-16
8-12	Example message preparation window	8-19

1.0 INTRODUCTION

1.1 BACKGROUND

The Navy is developing and fielding a number of computer-based systems to support command and control (C²) decision making by commanders and their watch teams on afloat platforms and in ashore facilities. Each system will be composed of a set of application programs that operate on a single workstation or are distributed among several workstations. Taken together, these applications will provide the range of functionality called for by the system. Some applications will be shared by multiple systems, while others will be unique to a single system.

It is critical to the overall usability of the Navy's C² systems that the applications being developed have a user interface with a common appearance and behavior. The user interface needs to be standardized so that each system will appear and operate in a consistent manner. In addition, users should be able to navigate easily both within and between applications in a single system as well as apply their experience with one application to other applications and other systems.

The user interface provides the link between users and an application and must be designed so that users can access and manipulate data easily, efficiently, and with minimal error. Interface standardization is particularly important for the Navy's C² systems as users are provided the capability to interact with a variety of complex, multi-windowed applications within a single system. The benefits to be gained from standardization are increased user productivity, reduced training requirements, improved system reliability, and increased efficiency in the development of individual applications as well as entire systems.

1.2 PURPOSE

This document defines a common appearance and behavior for the user interface of applications developed for Navy C² systems. The specifications presented here identify the types of objects users will see and the actions users will take to interact with these objects. The document focuses on the appearance and behavior of the user interface and does not address environmental specifications in areas such as lighting and noise, or ergonomic specifications in areas such as workstation design and layout.¹

¹ Although this document does not contain environmental criteria with which systems must comply, the specifications presented here have been developed to be appropriate for the range of environmental conditions (e.g., illumination) in which the systems will be installed.

The objective of the specifications is to establish a common "look and feel" for the user interface of the Navy's C² systems. That is, the user interface should be designed so that when users see an object on the screen, they are able to recognize both the type of function they can perform with the object and the means to perform the function. For example, a typical function performed by users is to change the size of a window. When an interface has a common "look," users know that any window with a resize border can be resized and, conversely, any window without a resize border can never be resized. When an interface has a common "feel," users know that whenever they see a resize border on a window, they can change the size by dragging the window border and that whenever they can resize a window, they do so by following this same method.

1.3 ASSUMPTIONS

1.3.1 Hardware and Software Configuration

The user interface specifications presented in this document assume a common hardware and software architecture for the C² systems to which the specifications will be applied. The hardware configuration will be a Navy standard desktop tactical-support computer, with at least one color monitor, a keyboard, and a pointing device (such as a mouse or trackball) containing three buttons.² Although the hardware configuration includes both a pointing device and a keyboard, the primary means for user input is assumed to be the pointing device; these specifications do not require that users be able to interact with a system using only the keyboard.

The software configuration will include a POSIX-compliant operating system such as Unix, X Windows as the windowing system, and Motif as the graphical user interface.³ Application developers shall use Motif unless a strong justification can be made for the variation.

1.3.2 System Functionality

The C² systems complying with these specifications are expected to provide functionality in the areas of information transfer, information management, and

²The specifications presented here are based on a Sun 4 Series computer with a Hitachi 19-inch color graphics monitor with 1280 x 1024-pixel resolution. Subsequent versions of this document will address other hardware configurations, including those with monochrome monitors, as well as new hardware configurations as they become available.

³These specifications are consistent with the Motif Style Guide, revision 1.1, except where noted.

tactical display and control.⁴ Each system will implement the functionality that is specified for that system. Users will be supported as they perform tasks in each of these areas by a set of application programs that perform a particular function or collection of functions within a system. The applications may be available within a single workstation or distributed among several workstations.

The C² systems will provide appropriately cleared users with access to a variety of military communications systems and with the capability to receive, store, and forward messages electronically. In addition, users will be able to access a tactical database containing technical, positional, and readiness information on fleet units. The database will be updated automatically with information from incoming message traffic. When necessary, users will be able to display and update information in the database and generate and release messages electronically. Users will also be provided with a real-time situation display of current tactical data that is updated automatically as message traffic is received. Users will be able to customize the display with specific overlays and to set filters so that only the tactical data of interest are presented in the display. Finally, users will be able to extract and integrate information from various sources and perform complex what-if analyses using the tactical aids and tools available within the system.

1.3.3 System Integration

It is assumed that the Navy's C² systems will use Government Off-the-Shelf (GOTS) software to provide the map display, tactical database manager, and incoming and outgoing message handling. Specific tactical functionality will be provided by a set of application programs that are appropriate to each individual system, and commercial off-the-shelf (COTS) software will be used as development tools and to provide general support in areas such as word processing, graphics, and spreadsheet. It is assumed that when new applications are developed, they will implement a Motif "look and feel" consistent with this document and that applications not using Motif will convert to this interface at an appropriate time during their development schedule. In addition, it is assumed that when COTS software is used, it will be configured, whenever possible, to be consistent with the specifications provided here.

Each C² system is expected to select the GOTS, application, and COTS software that, taken together, will provide the tactical functionality called for by the system. Although this document focuses on specifying the user interface for applications, it

⁴ The specifications focus on the user interface requirements of the command and control system operator and do not address these requirements for the intelligence analyst. A future version of the specifications will be expanded to incorporate these requirements.

also addresses aspects of the interface that need to be defined at the system level. It is assumed that each system will provide a top-level shell to control and manage access to applications and enforce consistency in the user interface for the system as a whole. This document identifies a number of system-level decisions that must be made with respect to user interface design and refers to the entity who has responsibility for these decisions as the system integrator for the system.⁵

1.3.4 System Users

The specifications presented here assume that each C² system will define the categories of users to whom system access will be provided. Each system will specify the functionality (i.e., specific applications) that will be available to each user category and control access to these applications during system login. For example, the functions available to a system administrator may be different than those available to normal users, and the functions available to normal users may, in turn, be configurable based on the tasks they perform.

These specifications assume that most users will have normal vision and hearing. However, the specifications define interface objects on the basis of a combination of features (not just color alone) and so are able to accommodate users with limited color vision.

1.4 SCOPE

This document was developed in accordance with the February 1992 version of the Department of Defense (DoD) Human-Computer Interface Style Guide.⁶ The DoD style guide provides guidelines for user interface standardization across applications developed for DoD organizations and calls for each organization to detail the specific "look and feel" for its applications in appendices to the guide. The user interface specifications presented in the current document serve as the appendix for the Navy's C² systems.

The user interface specifications presented here are consistent with the DoD style guide which presents guidelines for application development within layers 0 through 5 of the National Institute of Standards and Technology reference model (see figure 1-1). The specifications presented here provide additional detail

⁵ This document does not require that a system integrator be identified for each system. An integrator was defined as a convenience so that specifications could be provided for those aspects of the user interface that should be defined and managed at the system level.

⁶ Future versions of the specifications presented here will be revised as necessary to ensure continued compliance with the DoD style guide.

concerning layers 4 and 5 of the reference model as well as address layer 6 concerning application functions required by users. In addition, the specifications comply with the draft IEEE user interface drivability recommendations and are consistent with Section 15 on user-computer interfaces in Military Standard (MIL-STD) 1472D, with MIL-STD-1801, and with Military Handbook (MIL-HDBK) 761A. Finally, the specifications are based on published user interface guidelines (such as those from Apple Computers) and parallel the style guides for tactical systems prepared by the Air Force and Army.

	MODEL LAYER	SYSTEM COMPONENTS
6	Application	Application program
5	Dialogue	User Interface Design Language, User Interface Management System
4	Presentation	User Interface Design Language, User Interface Management System
3	Toolkit	Toolkit
2	Subroutine Foundation	Library functions
1	Data Stream Interface	Library functions
0	Data Stream Encoding	Network protocol

Figure 1-1. National Institute of Standards and Technology user interface system reference model.

In general, the specifications in the current document implement the Motif "look and feel" as detailed in the Motif Style Guide, revision 1.1. Deviations from the style guide were made as needed to accommodate operational requirements or constraints, provided that the deviations were consistent with established user interface guidelines, and to maintain consistency with military standards and IEEE recommended practice. When deviations are made, they are indicated in footnotes to the specifications.

1.5 INTENDED AUDIENCE

This document was written primarily for the user interface designers and software engineers who will create and maintain application programs for Navy C² systems. This document is also intended for the system integrators who will design and maintain the system-level user interface for these systems. It is assumed that this audience is familiar with the basic "look and feel" of the Motif graphical user interface and has identified the intended user population and the range of tasks they will perform with an application. The specifications presented here provide a design framework for this audience to follow so that the software programs they develop provide a user interface that is consistent at both application and system levels.

1.6 COMPLIANCE

System integrators shall develop the overall user interface for Navy C² systems in accordance with the specifications presented in this document. In addition, system integrators shall identify any system-unique user interface requirements and detail them in appendices to these specifications. Application developers shall comply with the specifications presented here as well as with the system-specific specifications for each system in which the application will be integrated. In addition, applications that make use of COTS development tools or integrate COTS software shall configure these products to be compliant with these specifications insofar as possible. Each application shall be reviewed prior to integration in a system by a government representative with expertise in user interface design to ensure compliance with the specifications.

The user interface specifications presented here shall be applied in the development of all new applications and in the upgrade of existing applications. Retrofit of existing applications shall not be required. Application developers shall comply with the version of the specifications in effect at the time the application is delivered.

The user interface specifications presented in this document shall be updated semi-annually. System integrators and application developers are responsible for recommending modifications to the specifications and for identifying system-specific specifications for individual systems. Requests from application developers for minor exceptions to the specifications must be approved by the developers of each system in which the application will be integrated. Requests from system or application developers for major exceptions to the specifications (e.g., to use a

graphical user interface other than Motif) must be approved by a Special Appeal Board consisting of representatives from each major Navy C² system.⁷

1.7 DOCUMENT OVERVIEW

This document begins by specifying the manner in which users interact with an application (section 2) and then defines the basic set of window, menu, and control objects available within the user interface (sections 3, 4, and 5). Detailed specifications are then provided on the overall design of system-level and application-level windows (sections 6 and 7) and on special design requirements for task-specific windows (section 8). Each section distinguishes those aspects of the user interface that need to be standardized and managed at the system level and those aspects that can be adapted at the application level to provide the functionality desired.

Throughout the document, specifications containing "shall" are mandatory provisions (which are required for compliance with the specifications), and specifications containing "should" are recommended provisions (to ensure consistency between applications).⁸ Finally, to support user interface standardization, the Motif resource settings that implement the specifications presented here and a checklist⁹ that contains the mandatory and recommended provisions of the document are included as appendices.

1.8 REFERENCES

The user interface specifications in this document were prepared after reviewing the following guidelines, standards, and style guides:

Air Force Intelligence Data Handling System (IDHS) Style Guide (Draft), October 1990.

Apple Computer, Inc. *Human Interface Guidelines: The Apple Desktop Interface*. Reading, MA: Addison-Wesley Publishing Co., 1987.

⁷ The charter and composition of the Special Appeal Board will be addressed in a future version of this document.

⁸ The distinction between mandatory and recommended provisions made in this document is generally consistent with the compliance conventions presented in the Motif Style Guide, revision 1.1.

⁹ The intent of the checklist is to provide a handy reference for developers as they implement these specifications in their applications. Because the checklist is not designed to be a standalone document, developers should consider it as a supplement to, rather than replacement for, the detailed provisions presented here.

Anti-Submarine Warfare Operations Center (ASWOC) Style Guide and Toolkit Specification (Draft), March 1991.

Department of Defense Human-Computer Interaction Style Guide, February 1992.

Department of Defense Intelligence Information System (DODIIS) Style Guide (Draft), June 1991.

Government Off-the-Shelf Software (GOTS) 1.0: Proposed Navy Tactical Command System (NTCS) Style Guide and Toolkit Specification, July 1990.

Human Factors Design Guidelines for the Army Tactical Command and Control System (ATCCS) Soldier-Machine Interface (Draft), Version 1.0, August 1990.

IEEE Recommended Practice for Graphical User Interface Drivability (Unapproved Draft 2), March 1992.

Kobara, S. *Visual Design with OSF/Motif*. Reading, MA: Addison-Wesley, Publishing Co., 1991.

Military Handbook 761A. Human Engineering Guidelines for Management Information Systems. 30 September 1989.

Military Standard 1472D. Human Engineering Design Criteria for Military Systems, Equipment, and Facilities. 14 March 1989.

Military Standard 1801 (Air Force). User/Computer Interface. 29 May 1987.

Open Software Foundation. *OSF/Motif Programmer's Guide. Revision 1.0*. Englewood Cliffs, NJ: Prentice Hall, 1990.

Open Software Foundation. *OSF/Motif Style Guide. Revision 1.0*. Englewood Cliffs, NJ: Prentice Hall, 1990.

Open Software Foundation. *OSF/Motif Style Guide. Revision 1.1*. Englewood Cliffs, NJ: Prentice Hall, 1991.

Open Software Foundation. *OSF/Motif User's Guide. Revision 1.0*. Englewood Cliffs, NJ: Prentice Hall, 1990.

Operations Support System (OSS) User's Manual. Release 1. 31 July 1991.

System integrators and application developers seeking standards for the design of workstations, associated furniture, and the facilities in which they are placed should consult MIL-STD-1472D and the following document:

American National Standards Institute. *National Standard for Human Factors Engineering of Visual Display Terminal Workstations*. Santa Monica, CA: The Human Factors Society, Inc., 1988.

2.0 USER-COMPUTER INTERACTION

2.1 INPUT DEVICES

Users shall make selections and control computer interaction by direct manipulation.¹ A pointing device (such as a trackball or mouse) shall be the primary means of interaction, allowing users to navigate on the screen and select objects for manipulation and action. The keyboard shall be used for text entry and shall be available for use along with the pointing device for performing frequently executed actions.

These specifications do not require that applications provide full interchangeability between keyboard and pointing device (so that users can switch between these input devices when performing any action). Instead, applications should be designed so that users interact with the input device(s) most appropriate to the action being performed. For example, if the action is track selection, user input should be with the pointing device; applications are not required to support track selection from the keyboard. If the actions performed in a window require different input devices, the window should be designed to minimize the amount of movement between input devices required to perform the actions.

Where appropriate, applications can support multiple input devices for performing the same action. For example, in a data entry task, users should be able to move the text insertion cursor between text entry fields from the keyboard or with the pointing device. If applications provide multiple input devices for a particular action, they should do so consistently so that the same device can be used whenever the action is performed. Finally, applications should be designed so that users can perform critical actions (such as saving data and quitting the application) with the keyboard as a backup in case of pointing device failure.

2.1.1 Pointing Device Input

2.1.1.1 The Pointer

The pointing device shall be associated with a single pointer that appears on the screen. The hotspot of the pointer (i.e., the active point) shall indicate the precise

¹ The focus on direct manipulation in these specifications deviates from the Motif Style Guide which supports both pointing device and keyboard-only interaction. Navy C² systems assume that the keyboard will be used for tasks such as text entry but not for tasks involving object selection and manipulation (e.g., plot control and track management). Hence, these specifications do not require total interchangeability between keyboard and pointing device.

location where operations occur. Users shall control the movement of the pointer by moving the pointing device; users shall be able to move the pointer anywhere on the screen. When users move the pointing device, the pointer shall move in the corresponding direction on the screen. For most user interactions, the pointing device-to-pointer movement ratio should be close to 1:1.

The position of the pointer should remain where it is placed on the screen until users move it; the pointer should not be moved arbitrarily by an application. MIL-STD-1801 recommends that the pointer should not deviate from its position by more than .05 inch in any direction; if a high degree of pointer stability is needed, the deviation should not be more than .01 inch in any direction. If more than one screen is available, the pointer shall move from one screen to another when users move the pointing device. However, the pointer shall not move beyond the outer screen boundaries or disappear from sight. The location of the hotspot shall not move on the screen as the pointer changes from one shape to another.

2.1.1.2 Pointer Shapes

The shape of the pointer depends on the functionality of the area on which the pointer is located. The pointer shapes listed in table 2-1 shall be used whenever an application provides the functionality indicated in the table.² Applications shall redefine the pointer shape only when the pointer is within an application window.³ The upper-left-pointing arrow shall be the general-purpose pointer for object selection in most windows. The X pointer shape, which is the default when running X Windows, shall not be used by an application since displaying this pointer means that users have access to all X (and hence operating system) functions. In the remainder of this document, the pointer is assumed to be an upper-left-pointing arrow, unless otherwise indicated.



Application developers shall not create a new pointer shape for functionality that already has a pointer associated with it, or use an existing shape to symbolize functionality it was not designed to represent. If necessary, developers can create a new pointer shape for functionality not listed in the table. The new shape should be easy to see (e.g., have high contrast with the background, not obscure other information on the screen), with a hotspot that is obvious and easy to locate. In

² The pointer shapes in the table are a subset of the ones defined in the Motif Style Guide (on pages 2-9 - 2-11) and are selected to provide the functionality that application developers are expected to need. Developers can use any of the Motif pointer shapes except the X pointer.



³ The management of the pointer shape in a multitasking environment (e.g., what application controls the pointer shape if multiple applications are running) will be addressed in a future version of this document.

Table 2-1. Pointer shapes.


Pointing

-  The upper-left pointing arrow is used in most window areas for object selection. The hotspot for the arrow pointer is the point of the arrow.
-  The I-beam pointer is used in text areas to position the text insertion cursor and perform actions on text. The I-beam pointer is hidden during the time between any keyboard action and pointer movement (i.e., when text entry is occurring at the location of the text insertion cursor). The hotspot for the I-beam pointer is on the vertical bar of the I-beam about one-third from the top.


Working/Caution

-  The hourglass pointer is used to indicate that an operation is being performed in a window area. When the hourglass pointer is displayed, all pointing device and keyboard actions are ignored in the area.
-  The caution pointer is used to indicate that action is expected in another window area before input can be made in the current area and that the pointer has no effect in the area. When the caution pointer is displayed, all pointing device and keyboard actions are ignored in the area.


Resizing

-  The resize pointers are used to indicate positions for area resize, with the direction of the arrow in the pointer indicating the direction of increasing size. The horizontal and vertical resize pointers indicate resize in either the horizontal or vertical direction. The diagonal resize pointers indicate resize in both the horizontal and vertical directions simultaneously. The hotspot for the resize pointers is on the elbow or the line at the position pointed to by the arrow. A resize pointer appears when the pointer is on the frame border.

Moving

-  The four-directional arrow pointer indicates a move operation in progress or a resize operation before the resize direction has been determined. During a resize operation, the four-directional arrow pointer indicates a direction for resizing and changes to the appropriate resize arrow when the pointer is on the frame border. The hotspot for the four-directional arrow pointer is the spot where the arrows intersect.

Sighting

-  The sighting pointer is used to make fine position selections (e.g., to select a location on a map display). The hotspot for the sighting pointer is the spot where the lines intersect.

addition, the new shape should provide a hint to its purpose and should not be easily confused with other objects on the screen.

2.1.1.3 Pointing Device Buttons

Each button on the pointing device shall have a specific function that is executed whenever users press the button. The Left button shall be used to select an object, and the Right button to display a pop-up menu. Each system should provide a user preference setting so that, if desired, users who are left-handed can change the function mapping to select objects with the Right button and display a pop-up menu with the Left button.

These specifications do not assign a specific function to the Middle button. If desired, a system may require that this button be used for a particular function (e.g., as another means to execute a default action). An application can map a function to the Middle button if the function does not contradict or interfere with the function assigned to this button by the system or by another application.

Users shall perform the following actions with the pointing device buttons:

Press	Depress and hold down a button.
Release	Release a button after it has been pressed.
Click	Press and release a button without moving the pointing device.
Double click	Press and release a button twice in rapid succession without moving the pointing device.
Move	Move the pointing device without pressing any buttons.
Drag	Move the pointing device while pressing a button.

2.1.2 Keyboard Input

2.1.2.1 Keyboard Functions

The keyboard should be used for entering text, moving the text insertion cursor, selecting menu options (via keyboard accelerators) and default push button actions, and performing a speed search in a list box (see section 5.5.3). Keystrokes made during text entry should appear at the location of the text insertion cursor. Keystrokes made in performing the other keyboard functions should not be displayed.

2.1.2.2 The Text Insertion Cursor

When a window has input focus (see section 3.2.1), the text insertion cursor shall indicate the area (e.g., text entry field) where text typed by users will be displayed.

The text insertion cursor shall appear in the first area where text entry is possible, at the beginning of whatever text appears in the area. The standard shape of the text insertion cursor shall be a flashing pipe (|). The flash rate for the text insertion cursor shall be 2-5 Hz. When a window loses input focus, the text insertion cursor shall remain visible but static (i.e., stop flashing). When a window regains focus, the text insertion cursor shall appear at the same location that it was when the window lost focus and shall start flashing.

When the pointer is in an area where text entry is possible, it should appear as an I-beam shape, as shown in figure 2-1. When users click the Left button in the area, the text insertion cursor should appear at the location of the pointer and the pointer should remain displayed. When users begin typing, the pointer should disappear from the screen. The pointer should reappear when users stop typing or when they move the pointing device.

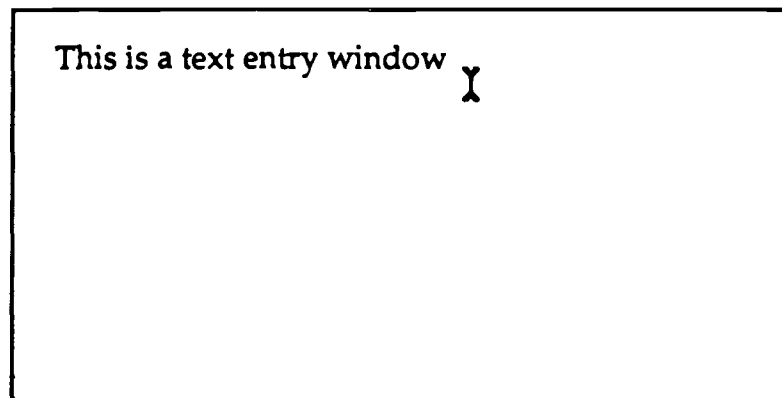


Figure 2-1. I-beam pointer in a text entry area.

Applications should be designed so that the pointer changes to an I-beam shape only when it is in an area where text entry is possible. In addition, applications should be designed so that the text insertion cursor can be placed only in these areas and cannot be placed in areas where text entry is not possible. Text entry should be possible only after the text insertion cursor is visible at a location that can accept text entry. Text entry should not be possible (i.e., should not be accepted by the application) when the text insertion cursor is not visible. Finally, applications should ensure that the text insertion cursor is highly visible whenever it appears in a text entry area. Visibility can be improved by changes in bolding, contrast, and size of the text insertion cursor.

2.1.2.3 Navigation Key Assignments

Applications shall provide key assignments for the standard navigation functions as defined in these specifications.⁴ Application developers seeking to assign keys to navigation functions not listed here should define these assignments in coordination with the system(s) in which the application will be integrated to ensure consistency in navigation functions within the system.

Because the specifications presented in this document do not require keyboard-only navigation, the key assignments listed below focus on the movement of the text insertion cursor within and between text entry fields containing one or more lines of text. Unless otherwise indicated, when a navigation function is performed that moves the text insertion cursor into a single-line text field, the cursor shall appear at the left margin of the field, preceding any text that is displayed in the field.

- a. **Forward and Backward Tabulation.** The Tab key shall move the text insertion cursor to the next text entry field when navigating between single-line text fields. The Tab key shall move the text insertion cursor to the next tab stop when navigating within multiple-line text fields. The direction of cursor movement shall be from left to right and top to bottom. When the text insertion cursor is at the bottom-rightmost field, pressing the Tab key shall move the cursor to the top-leftmost field. The Shift and Tab keys shall be used together to perform backward tabulation. This key combination shall move the text insertion cursor to the previous text entry field when navigating between single-line text fields, or to the previous tab stop when navigating within multiple-line text fields. The direction of cursor movement shall be from right to left and bottom to top within a window. When the text insertion cursor is at the top-leftmost field, pressing the Shift and Tab keys shall move the cursor to the bottom-rightmost field.
- b. **Text Insertion Cursor Movement.** The Down Arrow key shall move the text insertion cursor to the text entry field below the one currently containing the cursor when navigating between single-line text fields, or to the character position on the line below the current one when navigating within multiple-line text fields. Similarly, the Up Arrow key shall move the text insertion cursor to the choice above the current one, the Left Arrow key to the choice to the left, and the Right Arrow key to the choice to the right. In each case, when the text insertion cursor reaches the top-left, top-right, bottom-left, or bottom-right-most choice (e.g., text field or character), the cursor shall "wrap" to the appropriate choice based on the direction of movement associated with

⁴ The keyboard functions presented here are consistent with both the Motif Style Guide, revision 1.1, and the IEEE user interface drivability recommendations.

the arrow key. If necessary, scrolling shall be available so that the cursor remains visible when an arrow key is pressed. When the text insertion cursor is at the bottom of a scrollable area, pressing one of the arrow keys shall cause the area to scroll so that the cursor remains visible.

- c. **Forward and Backward Text Deletion.** When the Delete key is pressed in any text area, the character to the right of the text insertion cursor shall be deleted (forward deletion). When the Backspace key is pressed in any text area, the character to the left of the text insertion cursor shall be deleted (backward deletion). If more than one character has been selected, pressing one of these keys shall delete all of the characters selected.
- d. **Spacing.** When the text insertion cursor is in any text area, pressing the Spacebar shall insert a space character. When the text insertion cursor is in a multi-line text field, pressing the Enter key shall move the text insertion cursor to the beginning of the next line in the field (i.e., execute a carriage return).
- e. **Default Actions.** When the text insertion cursor is not in a multi-line text field, pressing the Return key shall invoke the default action if one is available in the window (see section 2.2.3).

2.1.2.4 Modifier Key Assignments

Modifier keys shall be used in combination with other keys or the pointing device to perform actions or invoke functions. The Shift key shall be used with the pointing device to perform object selection and deselection (see section 2.2.2), and the Control and Alt keys shall be used with the pointing device to drag objects to a new location (see section 2.2.4.1).

The Shift, Alternate, and Control keys, in combination with other keys, shall be used to define keyboard accelerators which allow users to select a menu item without using the pointing device. Table 4-1 lists the key combinations for the standard keyboard accelerators defined by Motif and required by the specifications presented here.

Application developers seeking to assign modifier keys not listed here should do so in coordination with the system(s) in which the application will be integrated to ensure consistency in these assignments within the system.

2.1.2.5 Function Key Assignments

These specifications require that the F1 function key shall provide access to context-specific help.⁵ Application developers are free to define other function key assignments as appropriate to the functionality being provided. However, developers seeking to assign function keys not listed here should do so in coordination with the system(s) in which the application will be integrated to ensure consistency in these assignments within the system.

2.1.3 Alternative Input Devices

Although these specifications assume a hardware configuration that includes a keyboard and pointing device, application developers are encouraged to use these input devices in unique ways that will support efficient user performance within the application. For example, improved hooking algorithms that allow users to select objects such as tracks quickly and easily (e.g., by moving the pointer near rather than on an object) might be implemented, and the keypad might be used as a set of variable action buttons, with the actions that are mapped to the buttons shown in graphic form on the screen. In addition, developers should determine if devices such as voice input, touchscreens, or lightpens are appropriate alternatives for user input and coordinate their implementation with the system(s) in which the application will be integrated.

2.2 OBJECT SELECTION AND MANIPULATION

2.2.1 Object-Action Selection

Applications should base their interactions with users on an object-action selection paradigm in which users first select an object of interest and then select an action to perform on that object. The objects may be text (e.g., individual characters, strings of characters or text, or items in a list) or graphic (e.g., track symbols on a map), and the action may require a single selection by users (e.g., select a command from a pull-down menu) or multiple selections (e.g., display a dialog window, select option(s) in the window, and select an action to execute).

⁵ This requirement is consistent with the IEEE user interface drivability recommendations.

2.2.2 Selecting Objects

The methods listed in table 2-2 shall be used, singly or in combination, to select and deselect text and graphics objects.⁶ In addition to these methods, users, where appropriate, should be able to quickly select and deselect all of the objects in a window, either as options in a pull-down menu or as push button actions.⁷ If users are required to select an object from a group of closely overlapping objects, applications should implement a method (e.g., blink coding) whereby users can quickly and accurately select the object desired.

2.2.3 Other Types of Selection

When a default action is assigned to an object, the action shall be executed by double clicking on the object. For example, double clicking on an icon shall restore it to a normal-size window, and double clicking on the Window Menu button in a window shall close the window. When a default action is available in a window, users shall execute it either by double clicking when making a selection in the window or by pressing the Return key after making a selection in the window. For example, when users double click on an item in a list (e.g., the name of a map), the item should be selected and the default action in the window (e.g., display the map) should be executed.

In addition, extended selection should be available for text objects. Users execute an extended selection by clicking on a text object multiple times. When extended selection is available, clicking once should place the text insertion cursor in the text, double clicking should select the word on which the cursor is placed, and triple clicking should select the entire line of text on which the cursor is placed.

2.2.4 Manipulating Objects

Users should be able to manipulate (e.g., move, copy, delete) the text and graphics object(s) they have selected either by (1) dragging them to a new location or (2) storing them temporarily in a clipboard buffer and then displaying them at a new location. Application developers should determine the types of objects to which each method applies and then provide that method whenever it is appropriate for the operation being performed by users. For example, an application may provide

⁶ The methods presented here are consistent with those provided in the IEEE user interface drivability recommendations and differ from some of the selection models defined in the Motif Style Guide, revision 1.1.

⁷ The methods presented in these specifications address selection and deselection of objects that appear in a single window. Methods for selecting objects that appear in different windows will be addressed in a future version of this document.

Table 2-2. Text and graphics object selection and deselection.

Selecting Text

To select one or more characters, position the pointer on the first character, press the Left button on the pointing device to start the selection, drag the pointer until it is on the last character, and release the button to complete the selection. The characters highlight as the pointer is dragged over them. If the pointer is dragged downward over several lines of text, the text to the right of the original pointer position highlights, and text to the left of the current pointer position also highlights.

To add contiguous text to an existing text selection, position the pointer on the last character in the text to be added to the selection, and press the Shift key and click the Left button on the pointing device. The text in the existing selection remains highlighted, and the text being added to the selection also highlights.

Deselecting Text

To deselect text, position the pointer outside the text and click the Left button on the pointing device. When text is deselected, it returns to its normal appearance. In addition, when new text is selected, the previously selected text is deselected.

To deselect one or more characters in contiguous text, position the pointer in the selected text, press the Shift key and click the Left button on the pointing device. The text from the current pointer position to the end of the selection returns to its normal appearance. The remainder of the text continues to be highlighted.

Selecting Graphics Objects

To select a single object, position the pointer on the object and click the Left button on the pointing device. When the object is selected, it highlights.

To select a group of contiguous objects, position the pointer on the first object, press the Left button on the pointing device to start the selection, drag the pointer until it is on the last object in the group, and release the button to complete the selection. As the pointer is dragged across the objects, a bounding box is displayed outlining the objects being selected. When the Left button is released, the box disappears and the objects that were in the box highlight.

To add an object to an existing selection, position the pointer on the object, press the Shift key and click the Left button on the pointing device. When the object is selected, it highlights. The object(s) in the existing selection remain highlighted throughout the selection process.

To add a group of contiguous objects to an existing selection, position the pointer on the first object, and press the Shift key and the Left button on the pointing device. Drag the pointer until it is on the last object in the group, and release the Shift key and Left button. As the pointer is dragged across the objects, a bounding box is displayed outlining the objects being selected. When the Shift key and Left button are released, the box disappears and the objects that were in the box highlight. The object(s) in the existing selection remain highlighted throughout the selection process.

Table 2-2. Text and graphics object selection and deselection (continued).

Deselecting Graphics Objects

To deselect an object, position the pointer outside the object (or on another object) and click the Left button on the pointing device. When an object is deselected, it returns to its normal appearance. In addition, when a new object is selected, the previously selected object is deselected.

To deselect an object in a group of selected objects, position the pointer on the object, press the Shift key and click the Left button on the pointing device. The deselected object returns to its normal appearance. The other objects in the group remain highlighted.

To deselect multiple contiguous objects in a group of selected objects, position the pointer on the first object, then press the Shift key and the Left button on the pointing device, and drag the pointer over the objects being deselected. A bounding box outlines the objects as the pointer is dragged across them. When the Shift key and Left button are released, the box disappears. The objects that were in the box return to their normal appearance. The other object(s) in the group remain highlighted throughout the deselection process.

text editing commands (e.g., Cut, Copy, Paste) when users are working in a word processing window but may consider the commands inappropriate to implement in the text entry fields within a data entry window (e.g., to prevent users from pasting text into a field that is not long enough to accept it).

Developers should ensure that access to these commands is provided in a consistent fashion (e.g., using the vocabulary and keyboard accelerators presented here) throughout the application. In addition, users should be able to view the contents of the clipboard buffer and should be informed (via message window) when they attempt to cut or copy an object whose size exceeds the capacity of the buffer provided by the application.

2.2.4.1 Dragging Objects to a New Location

Users should be able move or copy the object(s) they have selected by dragging them to a new location, as follows:

To copy an object to a new location, position the pointer on the object, then press the Control key and the Left button on the pointing device and drag the object to the new location. When the button is released, a copy of the object is pasted at the new location, and the original object remains highlighted and at its original location.

To move an object to a new location, position the pointer on the object, then press the Alt key and the Left button on the pointing device and drag the object to the new location. When the button is released, a copy of the object is pasted at the new location, and the original object is deleted.

These methods of manipulation should be used to transfer objects such as individual data records or entire files from one window to another. The windows may be in the same or different applications.

Applications should comply with the following rules for drag-and-drop operations provided by the Motif Style Guide, revision 1.1 (p. 3-16 - 3-17):⁹

Dragging from a noneditable component must be treated as a copy.

Dragging to a transport device, like a printer or mailbox, must be treated as a copy.

Dragging to a removal device, like a wastebasket or shredder, must be treated as a move.

Dragging text to or from a Text collection must be treated as a copy.

In other cases, the operation must be treated as a move.

2.2.4.2 Using Editing Commands

Users should also be able to manipulate text and graphics objects by executing the following set of editing commands:

Cut clears the clipboard, stores a copy of the object in the clipboard, and removes the object from the window. If the object is text, the remaining text is compressed (i.e., shifted to the left) to fill in the space. If the object is graphic, the space occupied by the object is left blank.

Copy clears the clipboard and stores a copy of the object in the clipboard. The object remains at its original location.

Paste copies the object in the clipboard to the new location. If the object is text, Paste copies the object to the current position of the text insertion cursor; the text appears to the left of the cursor and any existing text moves to the right.

⁹ These rules are consistent with those provided by the IEEE user interface drivability recommendations.

If the object is graphic, Paste copies the object to a location in the window with input focus. The object remains in the clipboard so that a copy of the object can be pasted whenever this command is selected.

Clear removes an object from a window without copying it to the clipboard and does not compress the space that the object occupied.

Delete removes an object from a window without copying it to the clipboard and compresses the space that the object occupied.

Undo reverses the most recently executed action (e.g., restores "cut" text to its original location).

These commands may be provided within an application as push buttons or as pull-down menu items (as shown in figure 2-2). The commands should be available to users to transfer text (e.g., strings of characters, blocks of text) and graphics (e.g., lines, shapes, entire figures) within a window and from one window to another in the same or different applications.

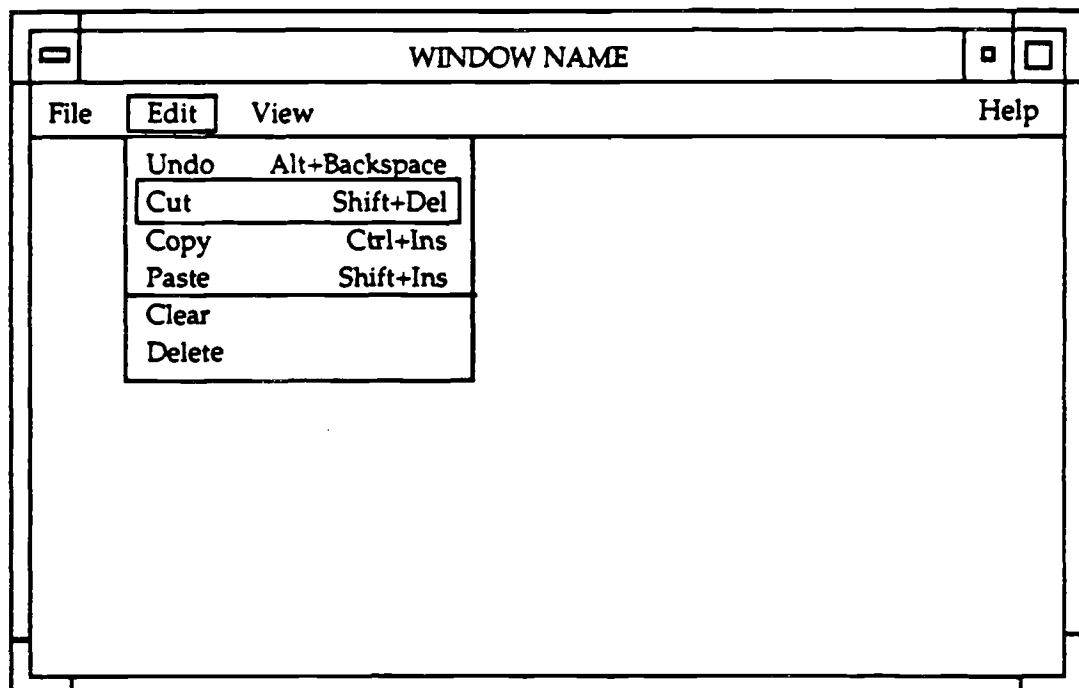


Figure 2-2. Example pull-down menu containing edit commands.

2.3 TEXT ENTRY

Applications should be designed so that insert mode is used for text entry.⁹ Clicking once should position the text insertion cursor at the location of the pointer. When users begin to type, the new text should be added at the insertion point, and the cursor and any existing text should move to the right. To delete the character to the left of the text insertion cursor (backward deletion), users should press the Backspace key; to delete the character to the right of the text insertion cursor (forward deletion), users should press the Delete key (see section 2.1.2.3).

Double clicking on text should select (and highlight) the word at the location of the pointer. In addition, dragging the pointer across text should also select (and highlight) it. In each case, when users begin to type, the highlighted text should disappear, the text insertion cursor should appear at the insertion point, the new text should be inserted at the insertion point, and the text insertion cursor (and any existing text) should move to the right. To delete the text that has been highlighted, users press the Delete key; the text should be deleted, the text insertion cursor should appear at the deletion point, and the remaining text should be compressed to fill in the space.

2.4 INTERACTIVE CONTROL

2.4.1 User Control of Interaction

Users should control the pace of the interaction with an application. The application should execute an action only in response to explicit user input. When appropriate, users should be able to specify when a process occurs (e.g., when a process should start or end or the rate at which a process occurs). In addition, users should be able to take actions that will interrupt or terminate a process.

2.4.2 Immediate Feedback

When users interact with an application, they should receive immediate feedback on the result of their action. Application developers should ensure that when users take an action (e.g., select a push button), there is an immediate and visible response to the action (e.g., the button highlights and the action indicated on the button is executed). A visible response to the action should occur even if the result of the action cannot be displayed immediately (e.g., the pointer should change

⁹ Insert (rather than overwrite) mode was selected to be consistent with the text entry procedures used in most commercial word processing applications since most Navy C² systems will include a word processing capability.

to an hourglass or a message should appear in the window while processing takes place).

An application should inform users when they can and cannot take an action. An application should provide visual cues indicating when it can accept input (e.g., the pointer appears as a shape supporting selection), when it is temporarily unavailable (e.g., the pointer appears as an hourglass), and when it is unavailable during extended processing (e.g., a progress message window appears on the screen). In addition, the appearance of an object should provide an indication of its availability. For example, the menu options in an application should provide feedback concerning the actions that can be performed on selected object(s). If a menu option is not appropriate for an object that has been selected or if a menu option cannot be executed for some reason, users should not be able to select the menu item and a visual cue (such as dimming the option) should be provided.

When users must perform several actions to complete an operation, an application should prompt users with the actions that need to be performed. For example, when users select a Zoom Area item from a pull-down menu, the application might provide feedback explaining how to select the area to be zoomed (e.g., a message at the bottom of the window indicating "Click on the map, then drag the pointer over the area to be zoomed.").

An application should ignore user actions made during periods when input cannot be accepted. For example, an application should accept keystrokes made by users only when the text insertion cursor appears in the application window with input focus; if the cursor is not present in the window, the keystrokes should be ignored by the application. In addition, an application should disable the pointing device and/or keyboard when input may have destructive effects (e.g., when the hourglass or caution pointer is displayed, indicating that the application is temporarily unavailable or that input cannot be accepted at the location of the pointer). Disabling is particularly important so that any input made by users during processing by the application is not stored and then executed when the application becomes available again. Although an application should not allow users to override disabling, users should be able to stop a process if desired (e.g., by selecting a Cancel button).

2.4.3 Response Time

Users should receive an immediate indication that their action has been accepted by the application. MIL-HDBK-761A recommends that some visible response should be made within .2 seconds of any user action (e.g., a checked button is highlighted, a push button is highlighted) and the response to simple requests for data display should take no longer than .5 to 1.0 seconds. Responses to requests for

new displays may take from two to ten seconds, especially if the operation requires more complicated operations such as accessing different files or transforming data. The IEEE recommendations indicate that an application should provide some visible response to a user action within .1 seconds and should notify the user if more than 5 seconds will be needed to produce a complete response. These guidelines should be considered as recommended response times; it is expected that actual response times will be determined by factors such as the hardware configuration being used, the size of the track database being maintained, and the amount of communications processing being performed.

Error feedback (e.g., when a user attempts an action that is invalid) should be provided to users within two seconds of the time the error was detected. In addition, users should receive feedback indicating that their action is being processed by the application. When a user's request requires more than two seconds to process, an application should change the pointer shape to an hourglass. When a user's request requires more than five seconds to process, an application should display a message window stating that a lengthy operation is underway and providing an indication as to the progress of the operation being performed. This type of feedback informs users that the application is not responding because it is processing their previous input; as a result, users will not continue to make selections that are executed when the application becomes available again.

2.4.4 Error Detection

If users attempt to execute an invalid action, the application should not execute the action except to display a message describing the action that is invalid. If users make multiple errors within a single action, they should be notified of each occurrence of an error. The feedback should be immediate, be visual and/or auditory, and explain the nature of the error made. For example, if users attempt to enter alphabetic characters when only numbers are valid, the application should provide feedback (e.g., display a message window) that informs users of the error. If users repeat an error, the feedback should be different (perhaps auditory with diagnostics or help also provided) to show users that their attempted corrective action was, in fact, processed. Users should be required to correct only the invalid action and not to repeat the entire action sequence. After correcting the error, users should execute the same action (e.g., select a push button) for re-entry that was used for the original entry.

2.4.5 Explicit Destruction

Applications shall be designed so that users must confirm destructive actions (i.e., actions with irreversible negative consequences such as deleting a data file) before the action is executed by the application. Users shall have to respond to a

confirmation prompt whenever they select a destructive action that cannot otherwise be reversed or “undone.” If the destructive action applies to more than one object (e.g., multiple files), the prompt should list the objects and allow users to select the one(s) to which the action should be applied.

Users should not have to confirm the action when they close a window unless the action will cause significant data loss. In general, closing a primary window is a potentially destructive action because this type of window usually generates data that should not be lost as a result of such an action. Closing a secondary window is usually nondestructive because these windows are used for obtaining additional input from users that does not have to be saved.

2.4.6 General “Undo” Capability

Applications should be designed so that users can “undo” the selection or action most recently made unless the selection was one requiring explicit destruction. In addition to being able to “undo” commands such as cut, copy, and paste, users should be able to deselect objects, return an object to its prior state before an action was executed, and retrieve information that was removed from the screen. If applications cannot provide an “undo” capability, they should label irreversible actions as such and clearly separate actions that are reversible from those that are not.

2.5 CONSISTENCY IN PERFORMING OPERATIONS

2.5.1 Consistency at the Application Level

Application developers should identify the set of operations that are supported within the application and then specify the sequence of actions that users follow to perform each operation. This sequence should allow users to complete the operation rapidly and with a minimum number of actions. Developers should ensure that each operation can be performed by following its standard action sequence which is the same throughout the application. Developers can provide other approaches to performing an operation (e.g., to accommodate differences between novice and expert users) but should ensure that the standard sequence for the operation is always available to users.

When defining an action sequence, developers should ensure that users have to execute only those actions that are required to perform an operation. For example, if an application requires users to enter location information for a contact by clicking the position on a map, the application should record the position and use the information; users should not have to type the same position information in a later

part of its dialog with the application. If the system does not record the position selected, then users should not have to perform the action (i.e., click on a position).

2.5.2 Consistency at the System Level

Each system should identify the set of operations that are performed across multiple applications and then specify the manner in which these operations will be executed within the system. For example, a system might specify that whenever an application presents a set of records (e.g., tracks or messages) containing sortable fields, users will sort the records by clicking on the title of the data field (see section 5.5.5). Users should be able to sort records by executing this action whenever they perform this operation in any application within the system. Users should not have to perform the same operation in different ways depending on the application being accessed (e.g., by selecting a Sort push button or selecting the data field as an option in a pull-down menu).

3.0 WINDOWS

3.1 WINDOW COMPONENTS

A window is a rectangular area of the screen in which an application displays information and interacts with users. The standard window, shown in figure 3-1, consists of (1) a window frame containing a title bar, resize border, Window Menu button, and window control buttons and (2) an application or client area within the frame for an application to use. The title bar contains a Window Menu button at the left edge and Minimize and Maximize buttons at the right edge; the name of the window appears in the middle of the title bar. The application area can be partitioned into multiple work areas to present different kinds of information.

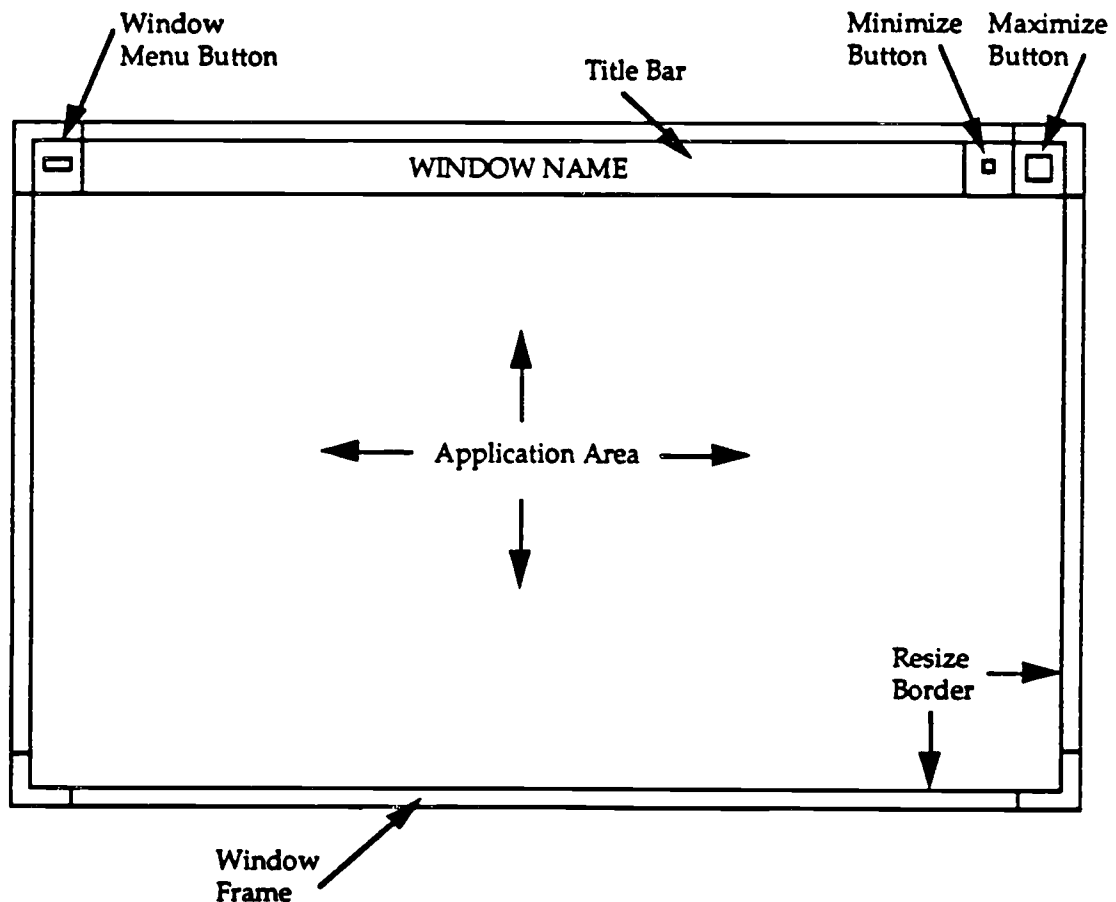


Figure 3-1. Standard components of a window.

A primary window shall contain all of the standard window components shown in figure 3-1. A secondary window shall contain a title area and a Window Menu

button. A secondary window may also contain resize borders and a Maximize button if users should be able to resize the window. A secondary window shall not have a Minimize button.¹

The Window Menu button in a primary window shall provide access to a menu containing the following window management options:

Restore	Restores a minimized or maximized window to its normal size. This function is not available (i.e., the item is dimmed) when the window is its normal size.
Move	Moves a window on the screen.
Size	Changes the size of the window in the direction indicated by the pointer.
Minimize	Changes a window into an icon.
Maximize	Enlarges a window to its maximum size.
Lower	Moves a window to the bottom of the window hierarchy.
Close	Closes a window and removes it from the screen.

The Window Menu in a secondary window shall contain the options corresponding to the window components included in the window. A secondary window shall not include a Minimize option in its Window Menu. If Lower is included in the Window Menu of a secondary window, selecting this option shall lower the window and all other windows that are children of the window's parent window (see section 3.4 on types of windows).

Application developers should ensure that each window in the application contains only those components that are appropriate for the window type and that the components in each window behave in a consistent manner. For example, selecting a Minimize button should always change the window into an icon. Each window should contain only one set of window components that are selectable by users; if an application incorporates windows that have been developed using a toolkit other than Motif, the application should integrate these windows (e.g., with a Motif window frame) so that only one set of components is available for window manipulation.

¹ The required components for primary and secondary windows presented here are consistent with the default window components specified in the Motif Style Guide, revision 1.1.

3.2 WINDOW STATES

3.2.1 Input Focus

Only one window on the screen shall have input focus (i.e., be able to receive input from the keyboard) at any time. A window shall have input focus when it first appears on the screen. An existing window shall receive input focus explicitly; i.e., users shall assign input focus by moving the pointer into the window and clicking the Left button on the pointing device.² If users click in an empty part of the window, it shall be raised to the front of the screen, and the window frame shall highlight. If users click on an object in the window, the window shall be raised to the front of the screen, the window frame shall highlight, and the object shall be selected. A window that loses input focus (i.e., when users click in a different window) may be obscured by other windows. When the window with input focus is closed, users shall explicitly select another window to have focus; an application shall not arbitrarily assign focus to another window on the screen unless emergency action is required.

3.2.2 Open, Active, and Closed Windows

A window is open when it is displayed on the screen. Multiple windows may be open at any one time and may partially or totally obscure one another on the screen.

An open window shall be considered to be active if processing is occurring in the window. The window with input focus is active until processing in the window is complete or the window is closed. A window without input focus may also be active if processing in it continues after it loses input focus. The content of an active window shall continue to be updated even if it does not have input focus.

When a window is closed, it shall be removed from the screen. In most cases, all processing relating to the window shall stop. Exceptions include windows where background processes, message transmission, and database updates are occurring.

3.2.3 Iconified Windows

An active window that does not need to be kept open may be iconified. When a window is iconified, it shall be replaced with an icon that is displayed along the lower left margin of the screen. Any processing that was occurring in the window shall continue while the window is iconified. When the icon is opened, the

² An explicit (vice implicit) input focus policy was considered to be appropriate for managing the processes running in multiple windows within a Navy C² system.

window shall be displayed at the front of the screen, at the same location where it was before being iconified.

3.3 WINDOW MANIPULATION

3.3.1 Placing a Window

Application developers shall define the normal, maximum, and minimum size of each window in the application and specify the location for each window when it first appears on the screen. Users may move and/or resize a window that includes these window management functions. When a window is closed and then reopened, it should be displayed at the same location where it was when it was last closed.

3.3.2 Moving a Window

Users shall be able to move a window using either of two methods. With the first method, users select the Move option in the Window Menu. The four-directional arrow pointer appears in the middle of the window. As users move the pointer, an outline of the window moves on the screen. Users click the Left button on the pointing device to place the window in its new location (and the pointer returns to its normal shape). With the second method, users position the pointer in the title bar, press the Left button on the pointing device, and then drag the window to its new location. An outline of the window moves on the screen as users move the pointer. Users release the Left button to place the window in its new location.

3.3.3 Resizing a Window

To expand a window to its full size, users shall either select the Maximize option in the Window Menu or click on the Maximize button in the title bar. In both cases, the window automatically expands to its full size.

To return a maximized window to its normal size, users shall either select the Restore option in the Window Menu or click on the Maximize button in the title bar. In both cases, the window automatically returns to its normal size.

Users shall be able to change the size of a window (not necessarily to its full size) using either of two methods. With the first method, users select the Size option in the Window Menu. The four-directional arrow pointer appears in the middle of the window. Users move the pointer to the frame border of the window. When the pointer reaches the border, it changes to a resize pointer. As users move the pointer, an outline of the window moves on the workspace. When users click the Left button on the pointing device, the window is redrawn to the new size (and the

pointer returns to its normal shape). With the second method, users position the pointer on the resize border, and the pointer changes to a resize pointer. Users then press the Left button on the pointing device and drag the border to change the window size. An outline of the window shall move on the screen as users move the pointer. The window is resized when users release the Left button on the pointing device (and the pointer returns to its normal shape).

3.3.4 Iconifying a Window

To iconify a window, users shall either select the Minimize option in the Window Menu or click on the Minimize button on the title bar. In both cases, the window is automatically iconified, and the icon shall appear in the lower left corner of the screen.³ If more than one window is iconified, the icons are arrayed on the screen in a row from left to right. The icon that is displayed shall contain a graphic image and a label (see figure 3-2). The label shall be located below the image and have the same title as its corresponding window.

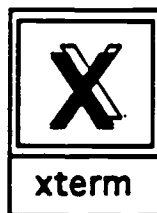


Figure 3-2. Example icon.

When a window is iconified, any ongoing processing associated with the window continues. If desired, icons can be moved using the same methods available for moving windows. To open an icon (i.e., restore a window that has been iconified), users position the pointer on the icon and double click the Left button on the pointing device.

3.3.5 Lowering a Window

To lower a window, users select the Lower option in the Window Menu. The window moves to the bottom of the window hierarchy on the screen.

³ Although the Motif Style Guide, revision 1.1 recommends the use of an icon box, the specifications presented here call for icons to be arrayed on the screen.

3.3.6 Closing a Window

To close a window, users shall either select the Close option from the Window Menu or double click on the Window Menu button. If processing is occurring or unsaved data have been generated in the window, users shall be required to confirm the action before the window is removed from the screen and processing stops.

3.4 TYPES OF WINDOWS

3.4.1 Primary Windows

Motif assumes that an application uses a primary window to conduct most of its interactions with users. When additional information is needed, a secondary window such as a dialog window⁴ is displayed to carry out context-specific dialogs. When the dialog is complete, the secondary window disappears from the screen. In figure 3-3, window 1 is a primary window, and windows 2 and 3 are secondary windows.

The specifications presented here assume that an application can have more than one primary window (e.g., a map primary window, a database primary window, a message generation primary window) depending on the functionality provided. The specifications also assume that an application may provide access to its functionality either directly (i.e., through a primary window such as a map) or indirectly from within another application (e.g., a tactical decision aid that is executed only when contact data are displayed on a map). In the latter case, the primary window for one application (i.e., the decision aid) may have the appearance of a secondary window in the map application.

Application developers shall define a hierarchy of primary and secondary windows that provide the functionality called for by the application. A family of windows consists of a primary (or parent) window and all of the secondary (or children) windows that are subordinate to the primary window. A primary window can have multiple secondary windows that are children to the parent window. Also, a secondary window may be the parent of one or more other secondary windows.

When a primary window is iconified, it and all of the secondary windows for which it is the parent shall be removed from the screen and replaced with an icon

⁴ These specifications refer to dialog windows and message windows (rather than dialog boxes and message boxes, as is done in the Motif Style Guide) to reduce possible confusion with window control objects such as list boxes.

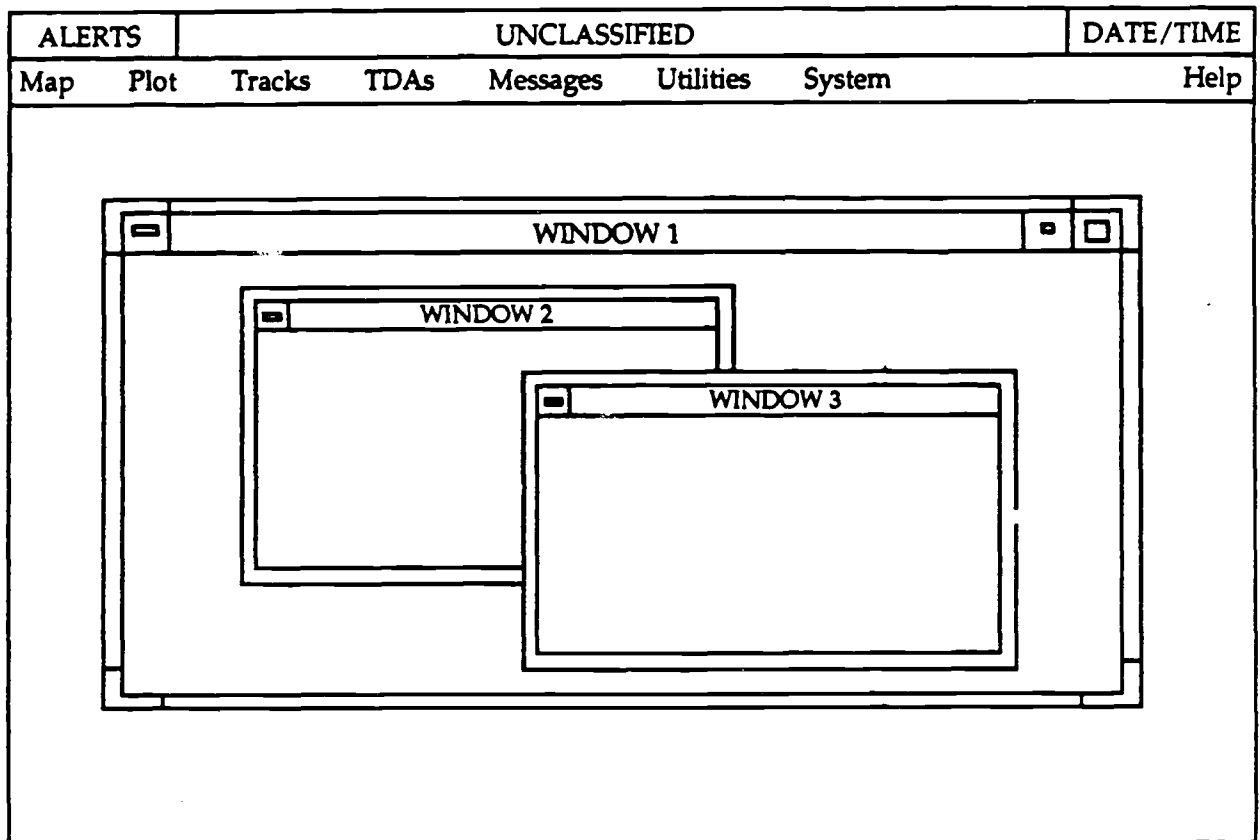


Figure 3-3. Example primary and secondary windows (from the Army guidelines).

representing the primary window. All processing in the primary window shall continue. When the icon for the window is opened, the window and all of its secondary windows that were displayed on the screen when the window was iconified shall re-appear. Each primary window in an application shall be iconifiable separately.

When a primary window is closed, it and all of the secondary windows for which it is the parent shall be removed from the screen, and all processing in the primary window shall stop (except as noted in section 3.2.2).⁵ When the last primary window for an application is closed, the application shall also be closed.

⁵ The behavior of a secondary window that is a child to multiple primary windows will be addressed in a future version of this document.

3.4.2 Secondary Windows

3.4.2.1 Dialog Windows

When a secondary window such as a dialog window is opened, it shall appear in front of its parent window, and the parent window shall remain displayed. A dialog window may be modeless or modal. If a dialog window is modeless, users shall be able to interact with other windows while it is displayed on the screen. If a dialog window is modal, users shall be restricted from interacting with other application and/or system windows while it is displayed. Motif defines three types of modal windows: primary modal (i.e., does not allow interaction with any parent of the window), application modal (i.e., does not allow interaction with any window created by the same application even if the application has multiple primary windows), or system modal (i.e., does not allow interaction with any other window on the screen). When a secondary window is closed, it and any other windows that are its children shall be closed but its parent window shall not be affected.

3.4.2.2 Message Windows

A message window is a secondary window that conveys noncritical information to users, displays progress in executing a lengthy process, or alerts users to an unusual event or potentially dangerous action. Users should be able to move a message window but not iconify or resize it. A message window may be modal or modeless depending on its purpose.

4.0 MENUS

4.1 PULL-DOWN MENU

A pull-down menu, shown in figure 4-1, shall consist of a menu title and a set of options or commands which are listed below the title and separated from it by a separator line. Each option should include a keyboard accelerator as appropriate. The title of each pull-down menu shall be displayed in a menu bar along the top margin of a window.

Edit	
Undo	Alt+Backspace
Cut	Shift+Del
Copy	Ctrl+Ins
Paste	Shift+Ins
Clear	
Delete	

Figure 4-1. Example pull-down menu.

4.1.1 Menu Title

The title of a menu should describe the category or type of options presented in the menu and should be different from the other titles that appear in the menu bar. The first letter of each word in the menu title should be capitalized. If the title contains an acronym, it should be capitalized.

4.1.2 Menu Options

4.1.2.1 Types of Options

Menu options shall be one of three types, shown in figure 4-2: (1) actions that are commands and are executed as soon as they are selected, (2) routings that display a window or a cascading menu, and (3) settings that are used to define parameters or specify an application state.

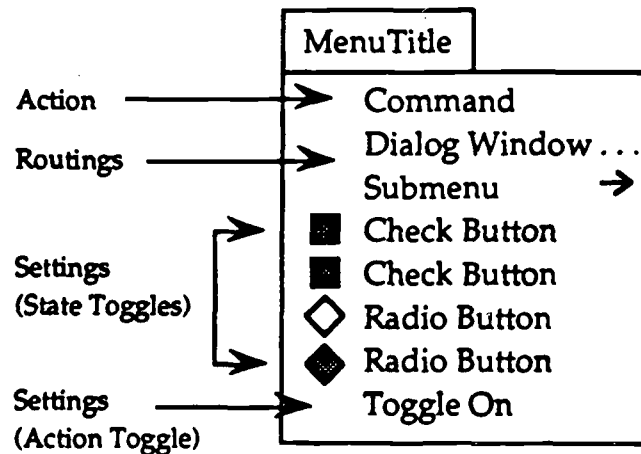


Figure 4-2. Examples of menu option types.

Routing options that display a window shall be identified in a menu by an ellipsis (i.e., they are followed by "..."). Routing options that display a cascading menu shall be followed by a right-pointing arrow.

Options that are settings may be action toggles (e.g., turn on/turn off) or state toggles (e.g., select a font size from the set available). Options that are action toggles should be worded to reflect the action that is implemented when the option is selected. The wording should be semantically congruent with natural usage; for example, if one toggle is worded "Move Object Up," the other toggle should be "Move Object Down," not "Move Object Back." When users select one of these options (e.g., "Turn On"), the wording of the option in the menu should change (e.g., "Turn Off") to reflect the action that will be implemented when the action is selected again. Only one of the action toggles shall appear in the menu at any time. If an "Undo" option is provided, the wording should change dynamically to reflect the action that can be undone. For example, if the most recently executed action is Cut, the option should be worded "Undo Cut."

Options that are state toggles should be worded to describe the state (e.g., a list of font names) and may be indicated by radio buttons (for exclusive settings) or check buttons (for nonexclusive settings), as appropriate, to the left of the option. When users select one of these options (by clicking on the menu option or the button), the check button or radio button should highlight but the option wording should not change.

4.1.2.2 Wording of Options

Each menu option should be phrased to reflect the action that is executed by the option (i.e., phrased as a command to the computer rather than as a question to the user), and worded in the vocabulary of users rather than that of application developers. The vocabulary listed in table 5-1 should be used if the actions listed in the table are included in the wording of menu options.

Each option should be tersely worded (preferably limited to a single word) and presented in upper- and lower-case letters, with the first letter of each word capitalized. All acronyms should be capitalized. Each option should be left-justified and appear on a single line; long menu options should be accommodated by making the menu wider rather than making the item take two (or more) lines.

4.1.2.3 Organization and Grouping

Developers that use pull-down menus in their applications should organize the options into logical or functional groups (as shown in figure 4-3), combine the groups into menus reflecting the categories of options addressed, and select titles that clearly describe each category. If the options cannot be organized into logical groups, they should be ordered according to frequency of usage within each menu, with the most frequently executed options at the top of the menu. Less frequently executed options and destructive options such as Delete and Clear should be at the bottom of the menu. If similar options are included in different menus, the options should be ordered in a consistent manner in each menu. If an organization based on logical groups or frequency of use is inappropriate, then the options may be ordered alphabetically or in numerical order.

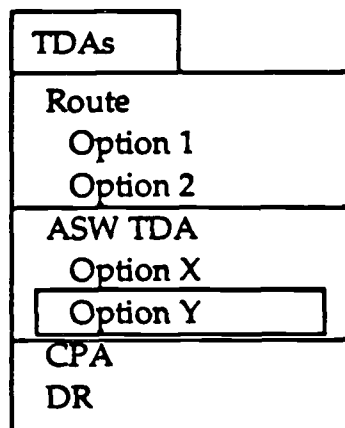


Figure 4-3. Example of logical ordering of menu options.

Whenever possible, a menu should contain no more than ten options. Menus with less than three options should be avoided. When a menu contains more than four options, the options should be divided into logical groups of no more than four options, and a separator line should be placed between the groups.

4.1.2.4 Available and Unavailable Options

If a menu option or set of options is never available to a user (e.g., system administrator commands), the option(s) should not be included in a menu. If an option is only temporarily unavailable, it should be displayed in the menu but dimmed (as in figure 4-4) to indicate that it cannot be selected. For example, an option may be unavailable because it cannot be executed in the window that currently has input focus. The wording of options may change (e.g., when an option is an action toggle). However, options should not be added to or deleted from a menu to indicate their availability within a particular part of an application.¹

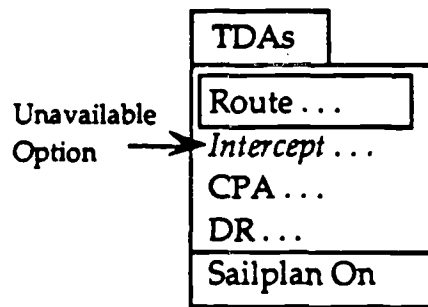


Figure 4-4. Example of an unavailable menu option.

4.1.3 Keyboard Accelerators

Applications shall use keyboard accelerators, shown in figure 4-5, to provide keyboard access to frequently executed menu options.² If an accelerator is available for a menu option, it should be right justified on the same line as the option and separated by enough space to make it visually distinct. Users shall type the keyboard accelerator for a menu option to execute the option. Users shall be able to execute the keyboard accelerators in the menu bar for the window with input focus. If a

¹ The inclusion of unavailable options in a menu is consistent with the Motif Style Guide but not with MIL-STD-1472D.

² Although mnemonics are available in Motif, the specifications presented here do not recommend their use by applications.

keyboard accelerator contains an alphabetic character, the accelerator should not be case sensitive. Users should be able to type the character in either upper or lower case.

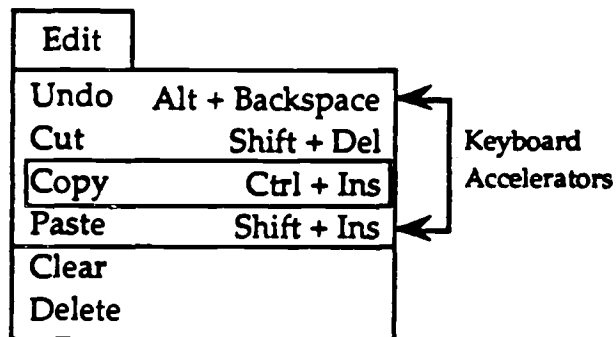


Figure 4-5. Example keyboard accelerators.

Application developers should implement the keyboard accelerators listed in table 4-1 for frequently executed menu options so that a standard set of accelerators are used by all applications within a system.

4.1.4 Cascading Menus

Cascading menus (see figure 4-6) should be used as needed to present groups of related actions hierarchically and shorten menus that are overly long (exceeding 10-15 options). The options in a cascading menu shall appear when users select or drag the pointer onto the parent menu option. The cascading menu shall be positioned either to the right of the parent menu or below the parent menu if space to the right is limited. The options in a cascading menu may themselves be the parent menu for another cascading menu.

The use of an excessive number of cascading menus should be avoided. Cascading menus should be limited to three levels and organized as shown in figure 4-7. If the number of multiple cascading menus becomes too large, new menus should be created or a dialog window should be used.

Table 4-1. Standard keyboard accelerators.³

<u>Action</u>	<u>Accelerator</u>
Close	Alt + F4
Copy	Ctrl + Ins
Cut	Shift + Del
Lower	Alt + F3
Maximize	Alt + F10
Minimize	Alt + F9
Move	Alt + F7
Paste	Shift + Ins
Restore	Alt + F5
Size	Alt + F8
Undo	Alt + Backspace

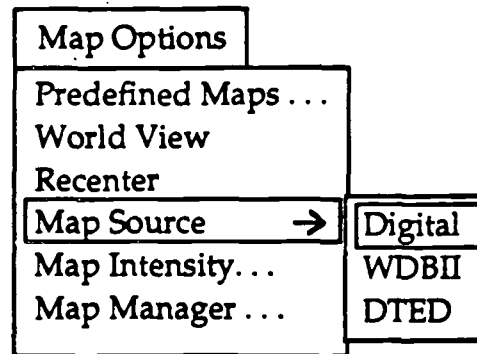
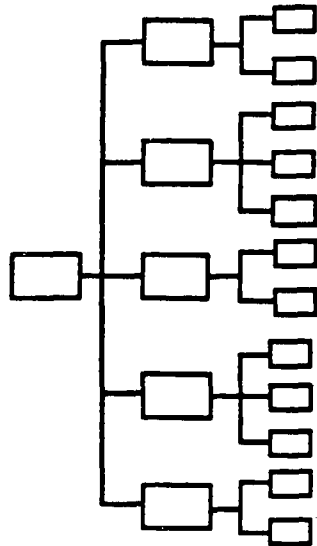


Figure 4-6. Example cascading menu.

³ These accelerators are taken from the Motif Style Guide; the set of standard accelerators to be used by applications will be expanded in subsequent versions of this document.

Do this:



Not this:

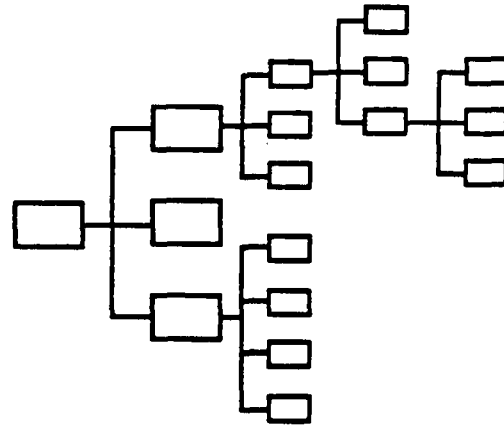


Figure 4-7. Organization of options in cascading menus (from the Army guidelines).

4.1.5 Menu Selection Methods

To browse the menus in a menu bar, users shall press the Left button on the pointing device and drag the pointer across the titles. Each menu shall appear as the pointer crosses its title. When a menu is displayed, the first available option in the menu shall highlight. Two methods shall be available for users to open a menu and select one of the options:

Drag Method - Position the pointer on the menu title. Press the Left button to display the menu, then drag the pointer to the menu option desired. The options in the menu highlight (and any cascading menus for which an option is a parent are displayed) as the pointer moves over them. Release the Left button to complete the selection. The menu disappears, and the action is executed. If no selection is desired, drag the pointer off the menu and release the Left button.

Click Method - Position the pointer on the menu title and click the Left button to display the menu. The menu is "sticky" (i.e., it remains displayed even though the Left button is not pressed), and the first available option in the menu highlights. Move the pointer to the menu option desired and click the Left button. The option highlights, the menu disappears, and the action is executed. If the option is a parent for a cascading menu, position the pointer on the option desired and click the Left button to select it. To "unstick" or

close the menu without making a selection, move the pointer off the menu and click the Left button. If users move the pointer off the menu and onto another object and then click the Left button, the object is selected and any action associated with the object is executed.⁴

Applications should include tear-off menus which remain displayed while users need to make multiple selections from the menu.⁵

4.2 POP-UP MENUS

A pop-up menu shall consist of a set of options and an optional title which appears above the options. A pop-up menu differs from a pull-down menu in that the title of the pop-up menu does not appear in a menu bar or anywhere on the screen. A pop-up menu can contain the same types of menu options and cascading menus as a pull-down menu and include keyboard accelerators as appropriate.

To display a pop-up menu, users shall press the Right button on the pointing device when the pointer is in an application window containing a pop-up menu. The menu shall disappear when users select an option (by releasing the Right button) or move the pointer off the menu without making a selection.

If pop-up menus are used in an application, they should be connected to the location of the pointer and appear on the screen near the object being manipulated. A window containing a pop-up menu should provide some indication that the menu is available (e.g., include text indicating the present of the menu in the window). In addition, if a pop-up menu is used, it should not contain options that duplicate other controls (e.g., OK and Cancel push buttons) that are already available in a window.

4.3 OPTION MENUS

An option menu, such as the one shown in figure 4-8, shall consist of a title and a set of options, only one of which is visible on the screen with the title. The title shall be a label (rather than a push button, as is the title of a pull-down menu) that appears next to or above the option. The option is a push button with a label containing the menu option that was most recently selected by users. The label shall

⁴ This behavior, which is considered optional in the Motif Style Guide, revision 1.1 is implemented in the specifications presented here to be consistent with the behavior of objects when input focus policy is explicit.

⁵ The appearance and behavior of tear-off menus will be specified in a future version of this document.

be followed by a bar graphic to distinguish the option menu button from an action push button (see section 5.1).

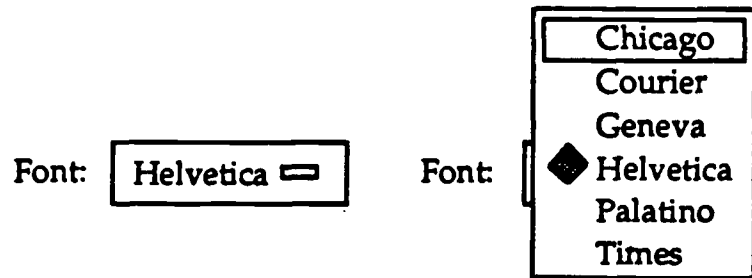


Figure 4-8. Example option menu.

When users position the pointer on the menu option that is displayed and press the Left button on the pointing device, the option menu shall appear. Users shall drag the pointer to the desired option and release the Left button on the pointing device to select the option. The menu shall disappear, and the option that was selected shall appear as the label in the option button.

5.0 CONTROLS

5.1 PUSH BUTTONS

A push button shall be used to initiate an action. A push button is a rectangular object containing either a short label describing the button's action or an icon representing the action that the button performs. Figure 5-1 presents examples of push buttons containing labels and action icons.

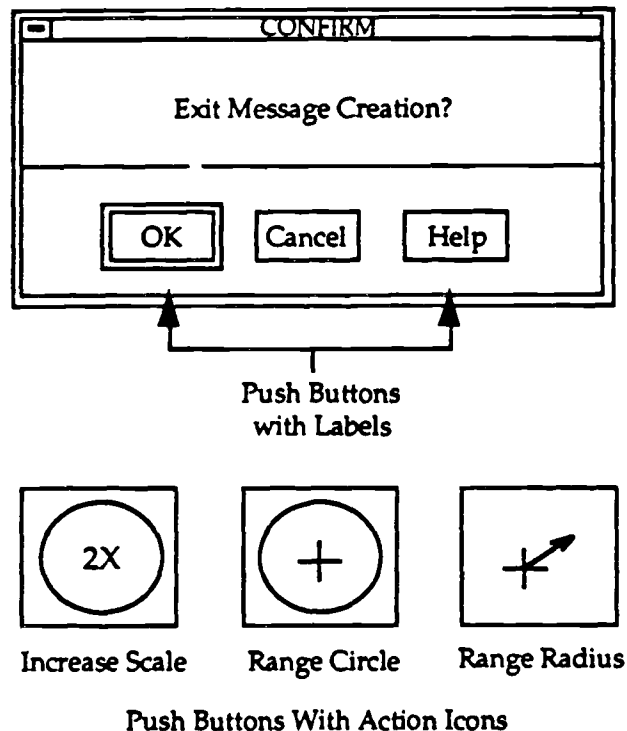


Figure 5-1. Example push buttons and action icons.

To select a push button, users shall position the pointer on the button and click the Left button on the pointing device. When the Left button is pressed, the push button shall highlight. When the Left button is released, the push button shall return to its normal appearance and the action represented by the push button shall be executed. To not execute the action, users shall move the pointer off the button while the Left button is pressed and then release the button. To execute the action represented by a default push button, users shall perform the operations described in Section 2.2.3.

5.1.1 Standard Vocabulary

Push button labels should be short and unambiguous. For action buttons, the label should describe the results of pressing the button and reflect the action that will be taken by the application rather than the user. For example, "Hook a Track" is not an appropriate label for a push button because it indicates the action that is executed by the user; this information is better presented as a message in the message bar of the window. When OK and Yes are used as response buttons, the question or statement that prompts the response should be worded carefully (e.g., phrased without negatives) and be unambiguous.

Terms such as "All" should be used in push button labels (e.g., Select All, Delete All) only when there is no ambiguity as to the object or data element to which "All" refers. If a push button label may have multiple referents within a window, the name of the object or data element should be used in the label instead of "All." For example, the access manager window shown in figure 5-2 contains two objects (i.e., a list of groups and a set of group assignments) and a number of push buttons, three of which include the term "All." Because this window contains multiple objects, it is important that users know what the referent is for the actions performed by these buttons (i.e., which objects do they affect) and whether the referent is the same for each button (i.e., do the buttons affect the different objects). Confusability can be reduced by renaming the buttons to indicate the object to which they refer, or by redesigning the window to locate the push buttons near the object they affect.

Application developers should use the vocabulary listed in table 5-1 to construct push button labels whenever applications perform the actions indicated in the table. Developers can create new vocabulary as needed to describe actions not listed in the table. If a new action is created, it should be a verb, stated in active voice, and describe the action that pressing the button causes. The names of actions should be congruent (e.g., Save/Delete, On/Off, In/Out). Developers should not create new vocabulary for actions already defined in the table, or use existing vocabulary to describe new actions. In addition, developers should ensure that the same vocabulary is used to describe the same action throughout the application.

5.1.2 Default Push Button Actions

Whenever appropriate, developers should identify a default action for each dialog window in the application and designate a default push button for this action. The default push button has an extra border around it (as in figure 5-3) and is activated when users press the Return key. Whenever possible, the default button should appear in the same position within the set of push buttons in a window. OK is frequently the default push button in dialog windows. OK and Yes should not be the label for the default button if the action executed by the button is potentially

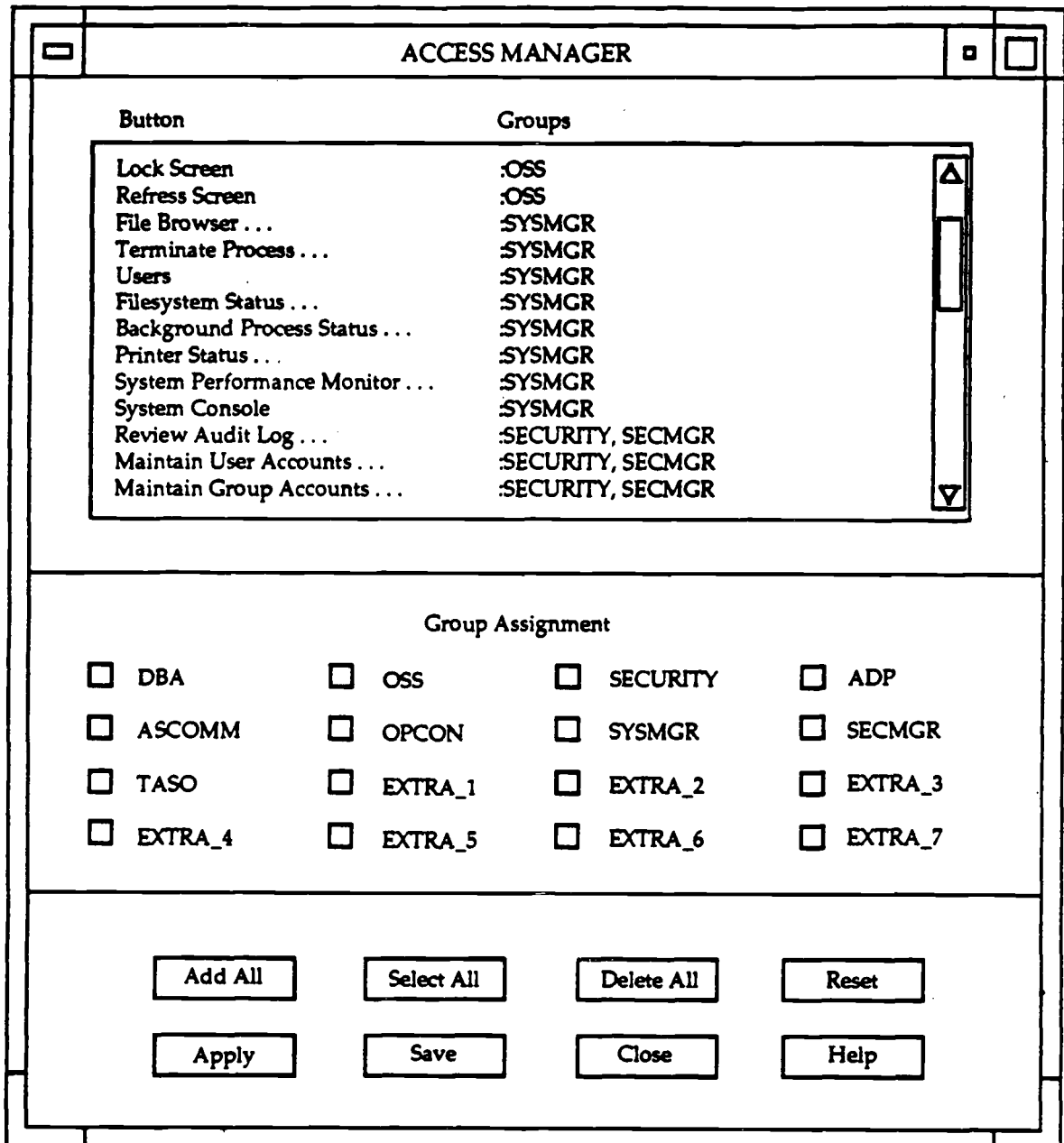


Figure 5-2. Access manager window.

Table 5-1. Action vocabulary.

<u>Term</u>	<u>Action Performed</u>
MANIPULATING PRIMARY WINDOWS	
These commands apply to windows where users can perform multiple actions (vice dialog windows where users usually perform a single action).	
Close	Ends processing occurring in a window and removes the window from the screen; requests confirmation if unsaved changes have been made and allows the user to save the changes. ¹
New	Opens a new window.
Open	Opens an already defined window.
Print	Initiates a process for printing the contents of a window (e.g., parts of a window, full window).
Save	Saves the contents of a window to a storage device.
MANIPULATING FILES AND DIRECTORIES	
These commands may be combined with the object of the command (e.g., Print File, Print Directory) if multiple commands and multiple objects are available in a window.	
Archive	Creates a backup copy of a file (e.g., on magnetic tape or fixed disk).
Close	Closes a file; requests confirmation if unsaved changes have been made to the file and allows the user to save the changes.
Delete	Removes a file from a storage device; requests confirmation prior to deletion.
New	Opens a new file.
Open	Opens an existing file.
Print	Initiates a process for printing the contents of a file.
Restore	Retrieves the backup copy of a file and saves it to a storage device.

¹ This definition of Close differs from that used in the Motif Style Guide, revision 1.1 which uses this term to "close the current primary window and its associated secondary windows" (p. 4-19) but uses Exit to "end the current application and all windows associated with it" (p. 4-20). The specifications presented here use a single term to simplify user interaction with windows on the screen since users do not need to distinguish the primary window that ends an application from the other primary windows in the application.

Table 5-1. Action vocabulary (continued).

<u>Term</u>	<u>Action Performed</u>
Revert	Replaces the current file with the version that was most recently saved.
Save	Saves a file to a storage device under the same file name; prompts the user for a name if the file does not have one.
Save As	Saves a copy of a file under a new name; prompts the user for the new name.
EDITING TEXT OR GRAPHICS OBJECTS	
Clear	Removes an object from a window without copying it to a clipboard buffer; does not compress the remaining space.
Copy	Duplicates an object to a clipboard buffer without deleting it from a window.
Cut	Removes an object from a window and stores it in a clipboard buffer.
Delete	Removes an object from a window without copying it to a clipboard buffer; compresses the remaining space.
Paste	Inserts an object from a clipboard buffer into a window at a selected location.
Undo	Returns an object to its state before the last operation was performed.
MANIPULATING ITEMS	
<p>"All" may be appended to these commands (e.g., Delete All) to indicate that the command applies to all items in a set of items.</p>	
Add	Adds a new item to a set of items.
Compare	Displays information on multiple items in a set of items.
Delete	Removes an item from a set of items.
Deselect All	Deselects (and removes highlight from) all items in a set of items.
Duplicate	Creates a copy of an item and adds it to a set of items.
Edit	Edits an item in a set of items.
Insert	Adds an item to a specific location within a set of items.
Mark	Annotates (e.g., with an asterisk) that an item has been selected.

Table 5-1. Action vocabulary (continued).

<u>Term</u>	<u>Action Performed</u>
Merge	Combines the contents of two items into a single item.
Next	Displays the contents of the next item in a set of items.
Select On	Allows the user to select a subset of items based on criteria specified by the user.
Select All	Selects (and highlights) all items in a set of items.
Sort	Sorts a set of items based on a variable selected by the user.
Transmit	Transmits an item via a communication channel selected by the user.
Unmark	Removes the annotation that an item has been selected.
View	Displays the contents of an item.
PAGING	
Next	Display the next page of information.
Previous	Display the previous page of information.
PROCESSING	
<p>"On" and "Off" may be appended to the name of a specific process to start and stop the process; for example, if "Monitor" is an automatic monitoring process, "Monitor On" starts the process and "Monitor Off" stops the process.</p>	
Continue	Resumes a process without changing data entries or control logic for the process.
Pause	Interrupts a process without changing data entries or control logic for the process.
Restart	Turns on a process that has been turned off; if the process is already turned on when the command is issued, the process is turned off, then turned back on.
Retry	Causes a process to be attempted again.
Start	Begins or turns on a process.
Stop	Ends or turns off a process at the next nondestructive breaking point.
Update	Checks the status of a process and displays the updated information.

Table 5-1. Action vocabulary (continued).

<u>Term</u>	<u>Action Performed</u>
EXECUTING CONTROL SETTINGS IN DIALOG WINDOWS	
Apply	Executes the control settings in a dialog window but does not close the window.
Cancel	Closes a dialog window without executing the control settings in the window.
OK	Executes the control settings in a dialog window and closes the window.
Revert	Restores the control settings in a dialog window to the values they had when the window was opened.
EXECUTING OTHER DIALOG WINDOW ACTIONS	
Help	Displays on-line information about an item or general information about a window.
No	Indicates a negative response to a question posed and removes the window containing the question.
Yes	Indicates an affirmative response to a question posed and removes the window containing the question.
SYSTEM-LEVEL COMMANDS	
Logout	Closes all windows on the screen and stops all processing by the system; requests confirmation if unsaved changes have been made and allows the user to save the changes.

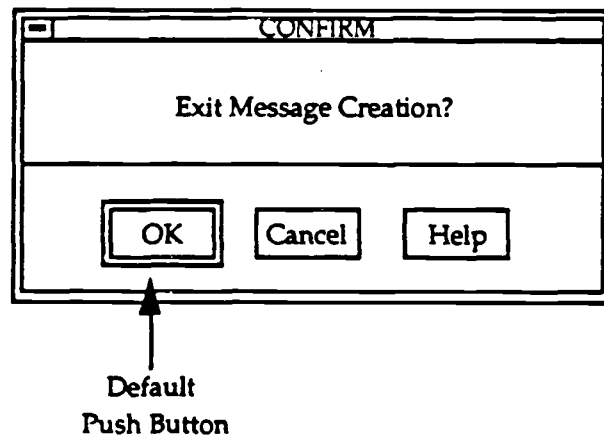


Figure 5-3. Example default push button.

destructive. If possible, applications should be designed so that the action performed by the default push button is reversible. If there is no default button in a window, pressing the Return key should have no effect, and users should have to select one of the available push buttons with the pointing device to execute an action.

5.1.3 Modifying Normal Push Button Behavior

The normal behavior for a push button is to highlight when the Left button on the pointing device is pressed and to return to its normal appearance when the Left button is released. Applications may modify this behavior so that a push button remains highlighted when the Left button is released and returns to its normal appearance only when another push button of the same type is selected. Applications should implement this modified behavior when push buttons are used both to execute an action and provide an indicator of the action selected. For example, when applications provide the capability to sort the contents of a multi-column list box (see section 5.5.5), the heading of each column should be represented as a push button. Users click on the heading to sort the contents of the list based on the items in the column, and the heading remains highlighted so that users can identify the column on which sorting occurred. When users click on another heading, the first heading returns to its normal appearance, the heading that was selected highlights, and the contents of the list are sorted on the items in that column.

5.2 RADIO BUTTONS

A radio button shall be used to select one option from a group of options. A radio button is a small diamond-shaped object, followed by a label describing the option represented by the button. Radio buttons are exclusive settings; they shall be used when selecting a single option from a set of at least two mutually exclusive options. Figure 5-4 presents an example of a group of radio buttons.

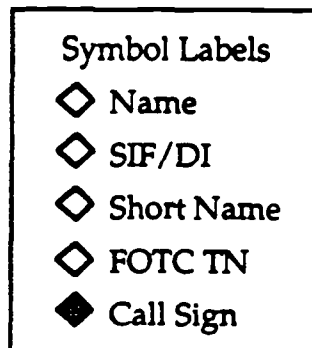


Figure 5-4. Example group of radio buttons.

To select a radio button, users shall position the pointer on the button or the label and click the Left button on the pointing device. The radio button shall highlight when selected. When a radio button is selected, any previously selected radio button in the group of buttons shall be deselected. Users shall deselect a radio button by either selecting it again or selecting a different radio button in the same group of buttons.

5.3 CHECK BUTTONS

A check button shall be used to set options in an application. A check button is a small square-shaped object, followed by a label describing the option represented by the button. A check button is a nonexclusive setting, with a function that is similar to a toggle switch; selecting a check button shall toggle to the setting or state indicated by the label but should not normally invoke any further action. Check buttons can be used singly or in related groups. Figure 5-5 shows an example of a group of check buttons.

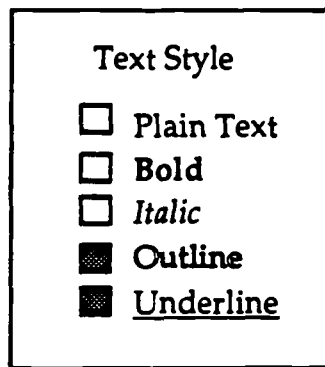


Figure 5-5. Example group of check buttons.

To select a check button, users shall position the pointer on the button or the label and click the Left button on the pointing device. The check button shall highlight when selected. When a check button is selected, any previously selected check buttons shall remain selected. Users shall be able to select any or all of the check buttons in a group. Users shall deselect a check button by selecting it again.

5.4 TEXT ENTRY FIELDS

A text entry field, shown in figure 5-6, shall be used to enter text. A text entry field shall include a title and a text entry area. The title shall appear either to the left or above the entry area and describe what is to be entered. The entry area may

include scroll bars if the text being entered is longer than the entry area or extends beyond a single line.

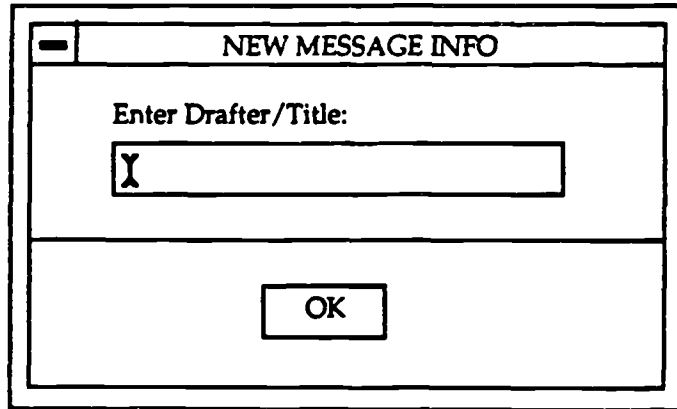


Figure 5-6. Example text entry field.

5.4.1 Format of Text Entry Fields

The title of a text entry field should include cues regarding format (e.g., Date [MM/DD/YY]) as appropriate. When a unit of measurement (e.g., ft, mi) is always associated with a field, it should be displayed as part of the title and not have to be entered by users. The title should be designed so that users can easily distinguish it from the text entry area.

The text entry area should indicate the basic features of the entry required. In addition, the entry area should be long enough for users to enter the information required. If the information being entered is a fixed length, then the entry area should be the same length as the information. If the information being entered varies in length, the entry area should be at least as long as the longest information. If window space is limited, an entry area may contain scroll bars. Finally, the entry area should include a decimal point if users are required to enter numeric data in decimal format.

The format of each text entry field should be consistent with users' expectations (e.g., users should type numbers as numerals and not as words). Whenever possible, the information called for in a field should be in "chunks" that are meaningful to users (e.g., whole words). If users have to enter a string of characters (that is not a word) longer than five to seven characters, the entry area should be broken into subgroups of three to four characters and separated by a space or delimiter, as shown in figure 5-7. Whenever possible, the subgroups should be

meaningful to users (e.g., day, month, and year in a date; the three parts of a Social Security number). Whenever possible, routine or default data, data already known by the application, or data that can be computed by the application should be automatically entered in a field. For example, if text entry fields are provided for the start date, end date, and duration of a mission, the user should have to enter only two of the values and the application should calculate the third.

Delimiters not required:	Delimiters required:	
Do this:	Do this:	Not this:
Day: <input type="text"/>	25/5/86	060589
Month: <input type="text"/>	25-12-90	91 05 07
Year: <input type="text"/>	15:3:92	7.5.90

Figure 5-7. Examples of text entry field delimiters (from MIL-HDBK-761A).

5.4.2 Distinguishing Text Entry Areas from Noneditable Text

Text that can be edited shall be displayed in a text entry field. Noneditable text (such as titles, labels, and directions) should normally be presented directly in the window and not be placed within a box so that it appears to be an object (e.g., a push button or a text entry field). However, if noneditable text must be displayed in a box, the box should have a different appearance (e.g., different color) so that it can be easily distinguished from a text entry area. In addition, when users move the pointer into an information-only area (i.e., a box containing noneditable text), the pointer should not change to an I-beam shape. When users click the Left button on the pointing device in an information-only area, the area should not change appearance and the text insertion cursor should not appear in the area.

5.4.3 Text Entry in a Text Entry Field

When a window containing one or more text entry fields is displayed on the screen, the text insertion cursor should appear in the first (i.e., upper leftmost) field in the window. If the field contains text, the text should be highlighted. Users should perform text entry as described in section 2.3. In addition, editing commands

such as Cut/Copy and Paste (see section 2.2.4.2) should be available if these functions are appropriate for the task being performed.

When users type variable-length information in a text entry field, the information should be automatically justified or truncated; users should not have to enter leading characters to fill the space available (see figure 5-8). In addition, users should be able to enter numeric data from either the keyboard or the numeric keypad, especially when users have substantial amounts of numeric data to enter. Finally, the amount of data users have to enter in a text entry field should be minimized; for example, users should be able to enter an abbreviation rather than an entire word and should not have to enter the unit of measurement associated with a number value.

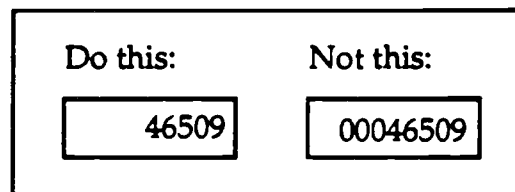


Figure 5-8. Example of automatic justification during text entry (from the Army guidelines).

5.4.4 Moving between Text Entry Fields

Users should move the text insertion cursor between text entry areas by pressing the Tab or arrow keys; applications should not automatically tab to the next entry area. When users press one of these keys to move the text insertion cursor to another text entry area, the cursor should appear at the left end of the area. In addition, if the text entry area already contains text, the text should highlight. Users should also be able to place the text insertion cursor in any of the text in a text entry area by positioning the pointer on the text and pressing the Left button on the pointing device. The size of the text insertion cursor should change as needed to match the size of the text font in which it is placed.

5.5 LIST BOXES

A list box such as the one shown in figure 5-9 shall be used to present a set of items from which to choose. The items in the list box shall be displayed vertically, with one item per line. The title of the list box shall describe its purpose or contents and appear above the box. A vertical scroll bar shall appear to the right of the items in the list box; users shall be able to move the slider in the scroll bar if the items in

the list exceed the space available in the box. The list box should scroll only in response to a user action (e.g., moving the slider on the scroll bar, conducting a speed or incremental search) and should not scroll automatically (e.g., whenever the content of a list is updated through an automatic process). A list box should be wide enough that users can read the items without having to scroll horizontally.

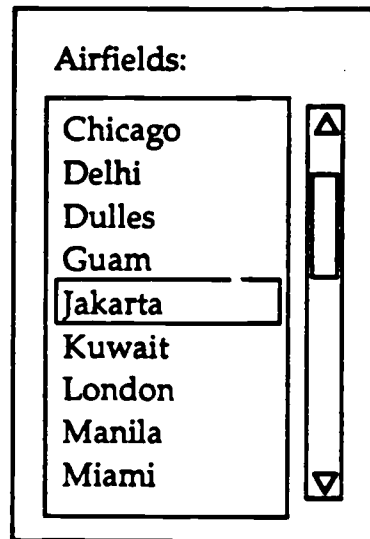


Figure 5-9. Example list box.

5.5.1 Ordering Items in a List Box

The items in a list box should be presented in sequential order based on the nature of the items and the sequence in which users expect the items to occur (e.g., chronological, alphabetical, sequential, functional, by importance). For example, a list of port names should be ordered alphabetically, and a list of messages by precedence, date-time group, or a combination of the two (e.g., date-time group within precedence).

5.5.2 Adding Items to a List Box

When users add an item to a list box, the item should appear in its correct position within the list (e.g., in numerical or alphabetical order) rather than at the end of the list.

5.5.3 Searching Items in a List Box

Users shall be able to search items in a list box by moving the slider on the scroll bar until the item appears in the box. In addition, applications should provide a speed search capability in all list boxes. If applications provide this capability, it shall function in the manner described here. When users type the first letter of the item being searched, the list shall scroll to the first instance of an item that begins with that letter. Users then use the scroll bars to scan through this part of the list and locate the item desired. The speed search capability shall be case-insensitive; i.e. users shall be able to type either an upper- or a lower-case letter to execute a search.

When the list box contains a large number of items (e.g., 50 or more), applications should provide a text entry field with the list box, as in figure 5-10, so that users can perform an incremental search. If applications provide this capability, it shall function in the manner described here. To perform this type of search, users shall type the first few letters of the item desired in the text box. When they press the Return key, the list scrolls to the first occurrence of an item that matches the letters. Users then use the scroll bars to scan through this part of the list to locate the item desired. If possible, an incremental search capability should be case-insensitive. However, if the search must be case sensitive, then this information should be provided to users, as in figure 5-11. If applications use a selection dialog box (such as the one in figure 5-12) to provide a search capability, the actions performed by users should be those described for an incremental search.²

5.5.4 Selecting Items in a List Box

To select an item in a list box, users shall position the pointer on the item and click the Left button on the pointing device. If the list box appears in a window that contains a default push button, users shall double click the Left button on the pointing device to choose an item and execute the default push button action. If more than one item in the list can be selected, users shall follow the methods described in table 2-2 to select multiple items.

² The Motif Style Guide, revision 1.1 recommends the use of this type of dialog box when users are required to make a selection from a set of items. The standard behavior required by users is to type the entire item in the text field provided in the dialog box. This behavior is inconsistent with the direct manipulation focus adopted in these specifications; requiring users to type the entire item assumes that they know the exact format of the item they want to select and can type the item without error. Providing an incremental search capability minimizes the amount of typing required by users and allows them to search for items without detailed knowledge about item format.

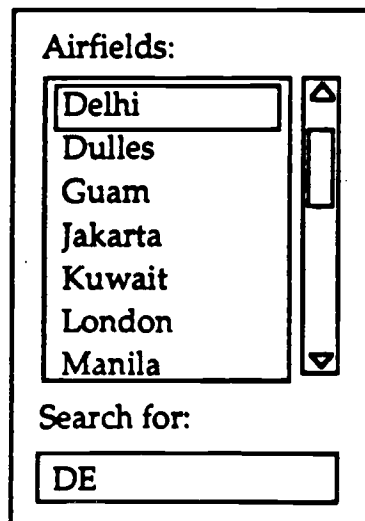


Figure 5-10. Example list and text entry field used in an incremental search.

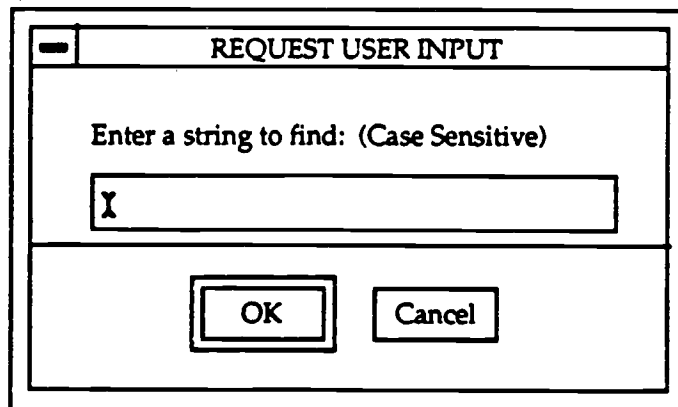


Figure 5-11. Example of a case-sensitive search.

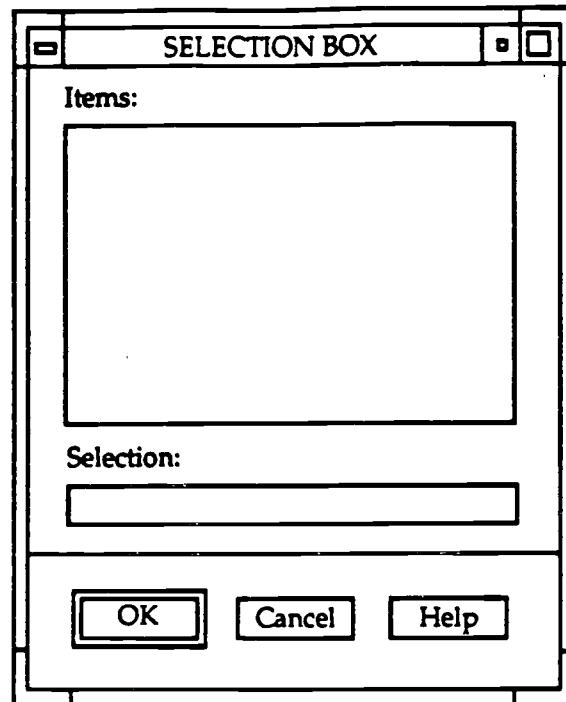


Figure 5-12. Example selection dialog box.

5.5.5 Sorting Items in a Multi-Column List Box

A series of list boxes may be arranged horizontally in a row to form a multi-column list box or matrix, as shown in figure 5-13. The list box titles serve as the column headings in the matrix, and the items in each list form the records that appear in the rows. Applications should provide a standard approach for sorting records that appear in this type of matrix. For example, users should be able to sort the records by selecting the column heading. With this approach, the headings of columns that can be sorted appear as push buttons, and when users click on the heading the records in the matrix are sorted in ascending order (e.g., numerical or alphabetical) based on the items in the column heading. The button for the heading remains highlighted (i.e., depressed) to indicate the column on which the records are sorted.

5.6 SCROLL BARS

Scroll bars shall be used to view textual or graphic information when it exceeds the space available to display it. Vertical scroll bars shall control backward and forward movement through the information; horizontal scroll bars shall control left and right movement.

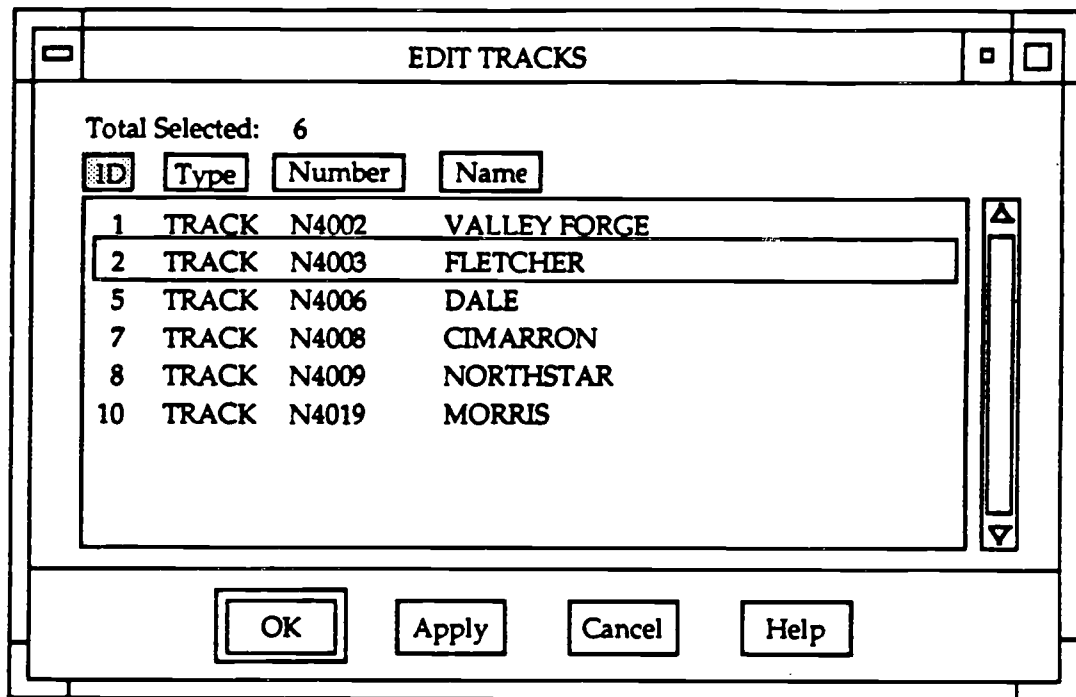


Figure 5-13. Example list matrix.

A scroll bar shall contain the components shown in Figure 5-14.³ The scroll region is the background of the scroll bar and shall represent visually the length of the information that users can scroll. The slider shall represent the window through which users view the information. The stepper arrows enable users to scroll incrementally through the information, and shall indicate the direction of the scrolling movement. The slider shall move back and forth in the scroll region and show the position of the currently displayed information relative to the entire amount. Users shall be able to scroll to the top or the bottom of the information but not beyond.

The relative position of the slider represents the relative position of the information currently displayed in the window. For example, the middle list box in figure 5-15 shows all of the items in the list so the slider fills the scroll region. In the other two list boxes, the position of the slider indicates that the items displayed are from the beginning of the list, and length of the slider indicates that the list is longer than the space available in the box.

³ Variations in the standard scroll bar (e.g., including dual-headed stepper arrows at both ends of the bar) will be addressed in a future version of this document.

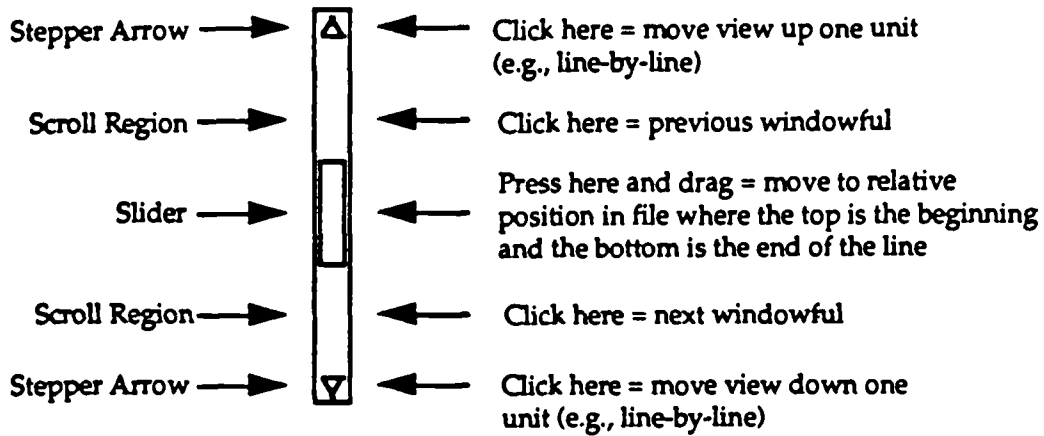


Figure 5-14. Components of a scroll bar (from the Air Force style guide).

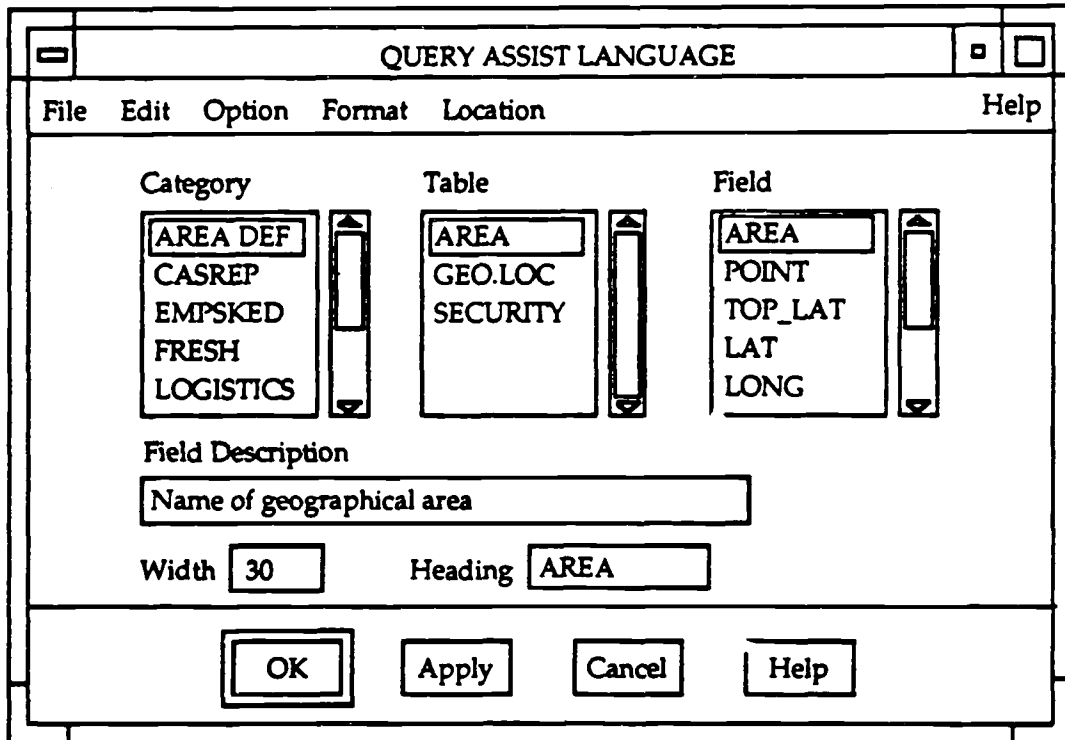


Figure 5-15. Examples of slider position within scroll regions.

When users click on a stepper arrow, the arrow shall highlight, and the window shall move through the information one unit (e.g., one line or column) at a time, in the direction indicated by the arrow. To move continuously through the information, users shall press on the arrow. When users click on the scroll region, the window shall move through the information, one window length (or width) minus one unit (for overlap) at a time. To move the window continuously through the information, users shall press on the scroll region. When users drag the slider, the slider shall move, and the window shall move to a location within the information that is consistent with the new slider location.

When users position the pointer in a scrollable window area and drag the pointer (i.e., with the Left button pressed) to the top or bottom of the area, the information should scroll in the direction that the pointer is being dragged. The information should scroll at the same speed as when users press on a stepper arrow. Scrolling should stop when users move the pointer back into the window area or outside the window or when users release the Left button on the pointing device.

5.7 SCALES

A scale (as shown in figure 5-16) shall be used to set or display a value in a range. Users shall display a value by adjusting a slider (bar or arrow) to a specified position along a line. A scale shall consist of a scale bar with tick marks representing the range of available values, a slider marking the currently chosen scale value, and a label above or next to the slider showing the current value of the scale. In addition, the scale should be labeled with the minimum and maximum values for the scale.

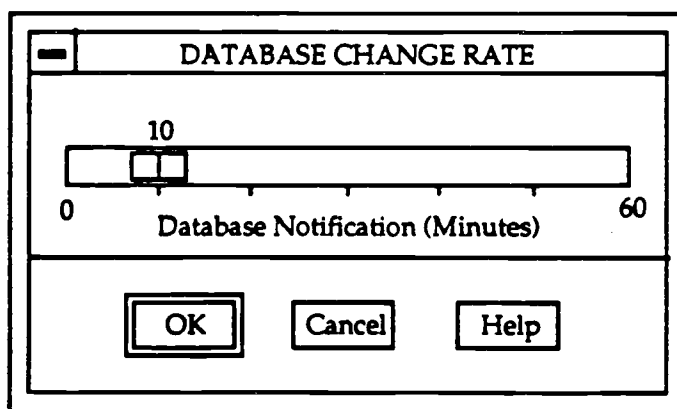


Figure 5-16. Example scale.

To display a value, users shall position the pointer on the part of the scale bar corresponding to the value desired and click the Left button on the pointing device. The slider shall move to this location and its value shall be displayed above or next to the slider. Alternatively, users shall position the pointer on the slider, press the Left button on the pointing device, and drag the slider along the scale bar. The value displayed in the label shall change as users drag the slider. When the value displayed corresponds to the value desired, users shall release the Left button.

6.0 SYSTEM-LEVEL WINDOWS

6.1 SYSTEM LOGIN

6.1.1 Login Procedure

Each system shall implement a login procedure that users must complete before they can access any system functions. A system may make available at login only those applications to which the user is allowed access, or may require users to log in to individual applications or groups of applications. If the system is unavailable for login, a message should be displayed, whenever possible, indicating the system status and when the system will become available.

6.1.2 The Login Window

A login window shall appear on the screen when users begin a session on a system. This window, shown in figure 6-1, shall contain two text entry fields for users to enter their user identification and password. The text entry areas shall be longer than the information that user will type in them so that their size will not give unauthorized users a clue as to the number of characters required for system access. The appearance and behavior of the objects in the login window (i.e., background, text entry fields, text) shall be consistent with the object specifications in table 7-1.

```
ASWOC SHELL
-----
Login: [ ]
Password: [ ]
```

Figure 6-1. Example login window.

Users must enter a valid identification and password before a session is initiated. If users enter an invalid identification or password, an error message shall appear in the window. Users who fail repeatedly to log on successfully shall be locked out of the system and informed that they should contact the system administrator.

6.1.3 Feedback during System Start-Up

System start-up occurs after users complete the login process successfully. During start-up, the system should display a message window indicating its unavailability, change the pointer to an hourglass shape, and disable the keyboard and pointing device. The message should disappear, the pointer should return to its standard shape, and the input devices should be enabled when start-up is complete and the system is available for input from users.

If appropriate, the system should provide messages indicating system status, e.g., average system response time (if system response is affected by the number of users) and known periods of unavailability.

6.2 THE SYSTEM WINDOW

6.2.1 Window Design

The system window shall appear on the screen when system start-up is complete. This window shall cover the entire screen, as shown in figure 6-2, so that users cannot access operating system functions. The system window shall contain a classification bar and a system menu bar, both of which shall extend across the top of the window; a title bar containing the system name may be included but is not required. The current classification level shall appear in the middle of the classification bar; if desired, a digital date-time clock may be displayed at the right margin of the classification bar, and various status indicators (such as alerts) may be included at the left margin. The system menu bar shall list the titles of the menus available at the system level; these menus shall provide access to the set of application programs available within the system.

The remainder of the system window shall be available for displaying application windows. If desired, action icons may be used to provide access to frequently executed actions within a system (e.g., starting/stopping processes, changing input/output devices). Action icons that are common to all applications within a system should be placed along the left margin (from the top of the window downward) of the screen. When users create application-level icons, the icons should be displayed in the lower left corner of the screen.

6.2.2 Window Appearance and Behavior

The system window should have the appearance indicated in table 6-1 and shown in figure 6-3. Appendix A contains the Motif resource settings (and RGB values) that implement the appearance defined in the table.

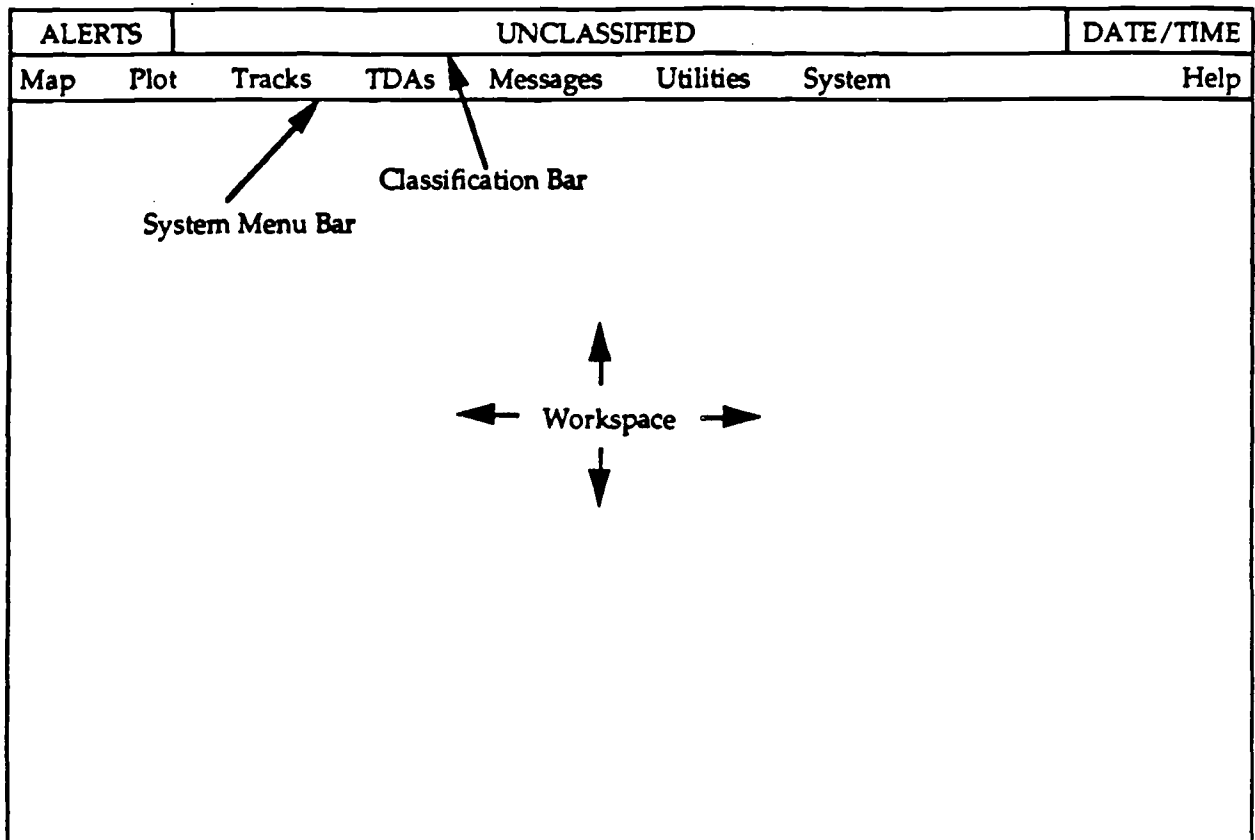


Figure 6-2. Example system window.

Table 6-1. Appearance (color and 3-D effect) of the system window.

<u>Object</u>	<u>Appearance and Behavior</u>
Classification Bar	Basic appearance: SkyBlue background, no 3-D effect, Black text
System Menu Bar	Basic appearance: SkyBlue background, raised effect, Black text When selected: Title appears as a push button with SkyBlue background, raised effect, Black text
System-Level Menu	Basic appearance when displayed: SkyBlue background, raised effect, Black text for available options, Gray text for unavailable options; first available option appears as a push button with SkyBlue background, raised effect, Black text When an option is selected: Option becomes a push button with SkyBlue background, raised effect, Black text
Workspace	Basic appearance: LightGray background, no 3-D effect

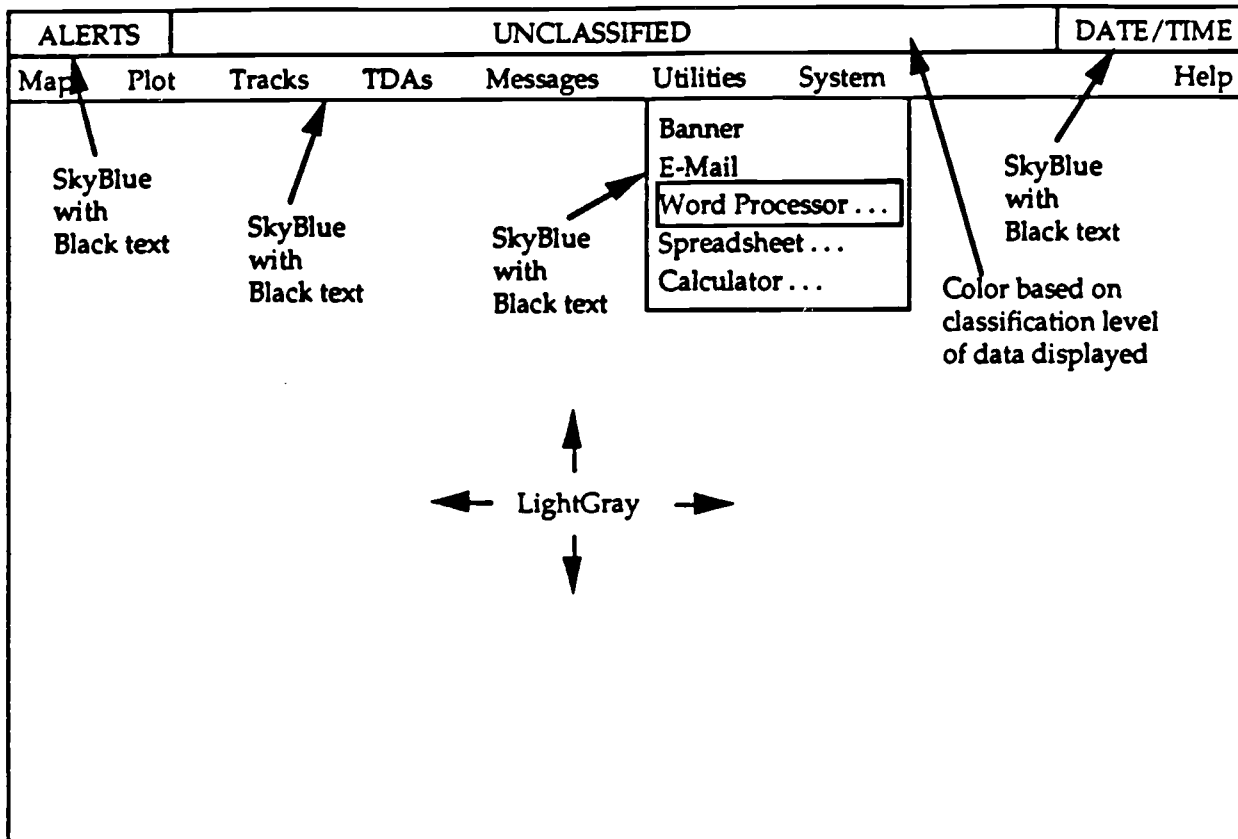


Figure 6-3. Example colors for the system window.

The system window shall not be moved or resized, and the classification and system menu bars shall not be obscured by application windows that appear on the available screen space. For example, in figure 6-4, Window 1 covers the entire screen but does not hide the system menu bar. The system window shall be active at all times so that users working in an application can select help or any other system-level menu options appropriate to the application.

Each system shall specify the initial appearance of the system window following start-up. For example, the screen may be empty, with users selecting the initial application with which to interact. Alternatively, the system may define a default application that is opened during system start-up, with a primary window for the application (e.g., a map window) appearing automatically on the screen. In this case, the selection of a default application should be determined by the system and may vary depending on the type of user (e.g., the system administrator may have a different default application than a normal user).

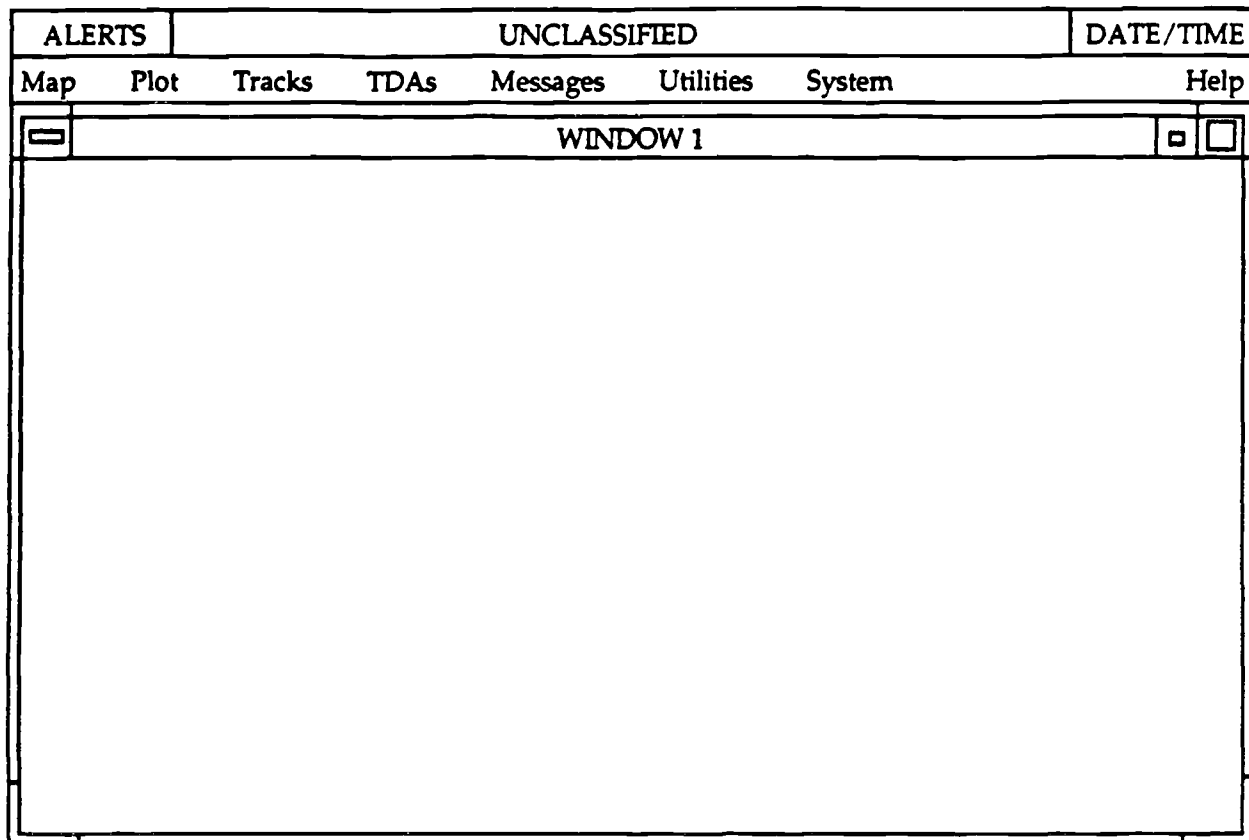


Figure 6-4. Maximum size of an application window.

6.2.3 Classification Markings

The classification marking indicated in the system window should be controlled by the system and reflect the highest level of data currently displayed on the screen. Each system shall implement a set of classification colors (with color names and RGB values), as follows:¹

- Unclassified = Green (ForestGreen; 34 139 34)
- Confidential = Blue (CornflowerBlue; 100 149 237)
- Secret = Red (Red; 255 0 0)
- Top Secret = Orange (DarkOrange; 255 140 0)

¹The colors listed here are taken from OPNAVINST 5510.1H; they are consistent with the classification bar colors for the Compartmented Mode Workstation listed in the DoD Human Computer Interface Style Guide.

These colors shall be used as the background in the classification portion of the classification bar. The text shall be all upper-case letters, and the text font and size shall be consistent with the specifications for text in windows presented in section 7.3.1. In addition, each system should provide an option for implementing alternate color sets as might be defined by the developing agency or the security officer at individual sites.

6.3 THE SYSTEM MENU

6.3.1 Integration of Applications into System-Level Menus

Each system shall determine whether the functionality provided by an application will be available as an entire menu (e.g., a set of map display options), as a group of options within a menu (e.g., specific message generation functions), or as a single menu option (e.g., a single decision aid or analysis tool). In addition, each system shall determine if an application shall be available whenever users select it from the system-level menu (e.g., map display) or only from within another application (e.g., a tactical decision aid that is executed only when contact data are displayed on a map).

6.3.2 Organization of the System Menu Bar

The set of menu titles that appear in the system menu bar should describe the overall functionality provided by the system. The menu bar should contain no more than ten menu titles plus Help. The titles should begin at the left margin of the menu bar and extend rightward, with enough space between them to be easily read and to accommodate the options in each menu.

In general, the leftmost menu titles should provide functions related to map and situation displays, the middle menu titles should provide functions dealing with database management, message handling, and tactical decision support, and the rightmost menu titles should provide system-related functions. Help should appear at the far right margin of the menu bar. Figure 6-2 is an example of set of menu titles that might be included in the system menu bar for a Navy C² system.

6.3.3 Consistency in Menu Titles between Systems

Each system should determine the extent to which it provides the same functionality as other Navy C² systems. Whenever possible, the same system-level menu titles should be used when accessing this functionality. For example, the following menu titles are for two systems that provide both common and unique functionality:

Map Tracks Database Messages TDAs Utilities System	Help
Map Tracks Environmental TDAs System	Help

When the same application is shared by multiple systems, it should have the same name in each system and should be available in the same system-level menu. For example, if a Fuel Calc application is one of the TDAs available in two systems, it should be named Fuel Calc in both systems and appear in the TDAs menu in both systems.

6.3.4 Access to Menu Options within an Application

Although each system defines the set of menu options that are included in its system menu bar, application developers should determine when these options are available for users to execute from within the application. When a system menu option is available within an application, the option should be displayed in normal text. Options that are inappropriate to execute from within the application should be dimmed. If all of the options in a menu are inappropriate, the menu title should also be dimmed.

Application developers should specify the number of instances of the application that can be running at one time. If only one instance can be running (e.g., on different screens), the menu option that launched the application should be dimmed while users are interacting with the application. If multiple instances can be running, the menu option should return to its normal appearance after it has been selected to indicate that users can select it again if desired.

When users are working in an application, they should be able to select Help from the system menu bar at any time. In addition, users should be able to browse all of the menu titles in the system menu bar even if all of the options in the menu are unavailable from within an application.

6.3.5 Quick Access to System Functions

Each system should include navigation aids such as that shown in figure 6-5 as an alternative to the set of function-oriented menus available in the system menu bar. These aids should be accessible through Help in the system window. The purpose of these aids is to provide an overview of system functions that can also be used to navigate quickly to a particular part of the system (e.g., move laterally between applications that may be available in different menus). Figure 6-5 presents a graphic overview of the functions available in an Army C² system. When users select one of the items in the graphic, they are automatically taken to that part of the system, and the appropriate window(s) are displayed on the screen.

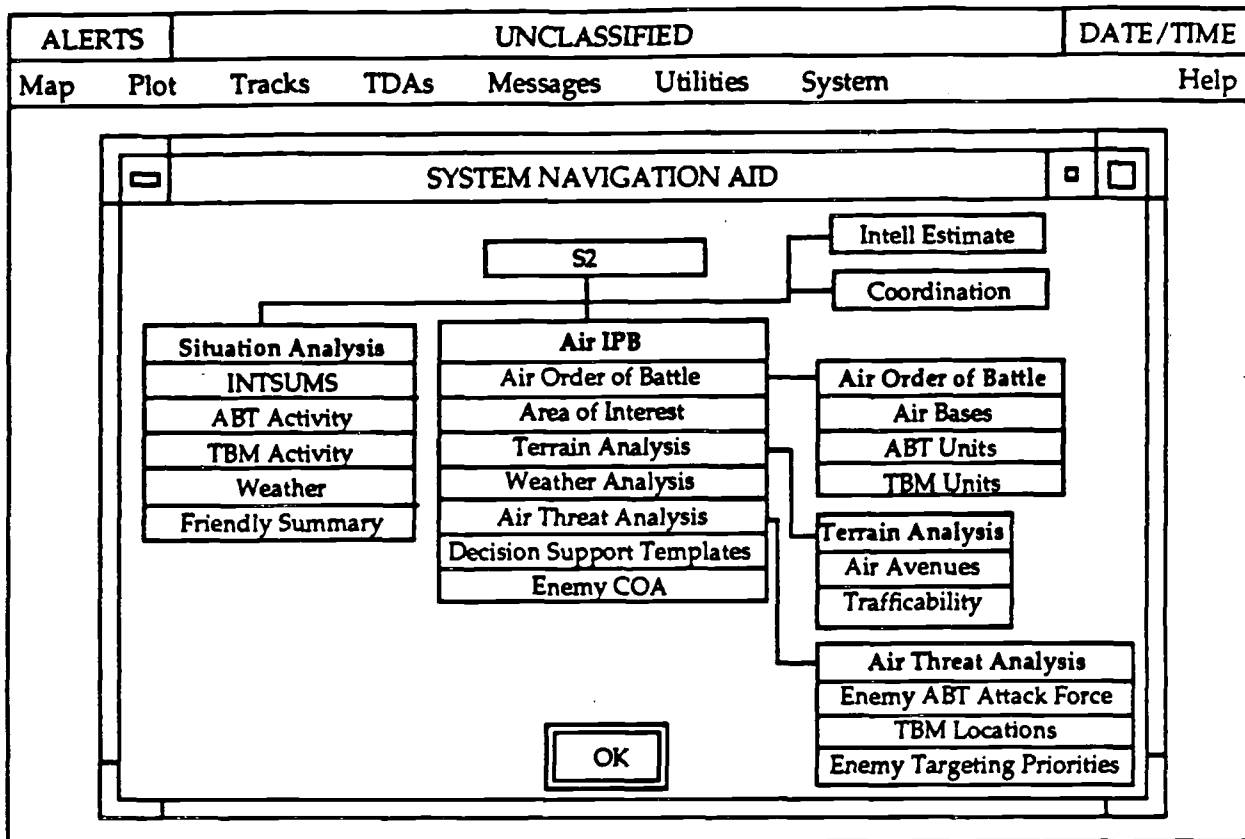


Figure 6-5. Example graphic navigation aid (from the Army guidelines).

6.4 SYSTEM SUPPORT

6.4.1 System Support Functions

Each system should include the resources and utilities required to support the overall tactical functionality provided by the system. This support should be provided in the Utilities and System menus in the system menu bar. The Utilities menu should include basic utilities (e.g., calendar, notepad, calculator), COTS software (e.g., word processing, spreadsheet, electronic mail), macros, and briefing presentation support. The System menu should contain options to end a session, print screens, review system status, define user preferences, manage alerts (e.g., set alert criteria, review alert logs), change a password, access peripherals, and perform file management.

Users should have the option of printing the entire screen or a specific window with and without its secondary windows. The system should provide the appropriate security marking (without the classification bar color) and produce an

accurate translation of color shading from the display to the hardcopy medium. In addition, users should be able to create, access, modify, and delete files and to do so without detailed knowledge of the file system structure. For example, windows for file manipulation might use list boxes for navigating between directories (so that users do not need to know full path names) and selecting directory and file names (so that users do not have to type them).

Each system should implement a screen saver (i.e., blank the screen if there is no user input for a period of time, e.g., five minutes) and provide users with the ability to rapidly suppress a display (i.e., quickly remove it from view to prevent the compromise of protected or classified data). In addition, each system should allow users to temporarily suspend a session without completely logging out. When a session is suspended, the system should continue all active processes but should not allow any interaction (i.e., accept any keystrokes) unless a user logs in to continue the session.

6.4.2 User-Specified Interface Settings

Each system should identify the set of interface parameters (e.g., functions assigned to pointing device buttons, criteria for alert generation, elapsed time before automatic logout) that users are allowed to set, given the tactical functionality provided by the system. The system should set a default value for each parameter that, unless changed by users, applies to all applications within the system. Some parameters may be in effect for the current session only, and some may be saved and implemented whenever the user logs in to the system. Users should be able to review the parameters and reset them at any time during a session. When users end the session, the settings in effect for the current session only should revert to the default values specified by the system.

6.5 SYSTEM LOGOUT

6.5.1 User-Initiated Logout

To end a session, users should select a Logout option from the System menu in the system menu bar. Users shall be prompted to confirm the action if any unsaved data have been generated and, if necessary, to save the data. Where appropriate, users should be prompted to log out of any applications to which they were required to log in. During the logout process, all processing in application windows shall stop, and all windows (including the system window) shall be removed from the screen. When logout is complete, the initial login window shall be displayed.

6.5.2 Automatic Logout

If a system implements automatic logout, a standard time (e.g., 15 minutes of user inactivity) should be designated by the system before automatic logout is initiated. This period should be modifiable by users as a user-defined parameter. The system should display a message during periods of inactivity indicating the action (e.g., keystroke, pointing device movement) needed to avoid automatic logout. In addition, an auditory signal should be presented at intervals during the period of inactivity. When automatic logout occurs, any unsaved data should be saved, and a message should be displayed indicating that automatic logout has occurred and providing the file name where any data have been saved.

7.0 APPLICATION-LEVEL WINDOWS

7.1 WINDOW ORGANIZATION

7.1.1 Primary and Dialog Windows

Primary and dialog windows have four major window areas, shown in figure 7-1: a window title, a window menu bar, a main area, and a message bar. All of the primary and dialog windows in an application shall contain a title and a main area; the inclusion of a window menu bar and a message bar depends on the type of window and the functionality it provides.

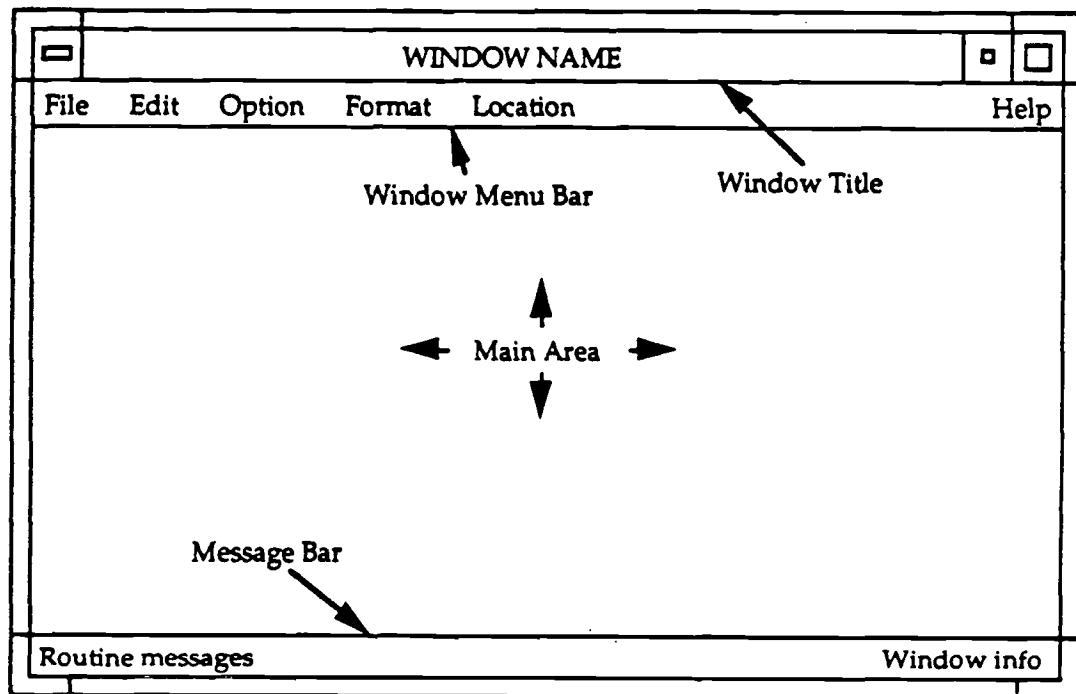


Figure 7-1. Recommended format for primary and dialog windows.

7.1.1.1 Window Title

A window title shall appear in the title bar of the window. The title shall be centered in the title bar and presented in upper-case letters. Each window title in an application should be unique. If an application contains multiple primary windows, the title should indicate the purpose of the window. The title of a dialog window should also describe its purpose. If users display a dialog window by selecting an

option in a pull-down menu, the title of the window should match the wording of the option.

7.1.1.2 Window Menu Bar

A window may include a menu bar containing pull-down menus that provide access to application functions. A menu bar is more likely to be part of the primary window(s) of an application but may also be included in a dialog window. If a window includes a menu bar, it shall appear below the title bar. The menu bar should contain no more than ten menu titles plus Help. The titles should begin at the left margin of the menu bar and extend rightward, with Help available at the right margin of the menu bar.

The options in a window menu bar should be unique to the application and not duplicate functionality that is available in the system menu bar. For example, an application that performs a what-if analysis on contact data on a map display should access map and plot control from the system menu bar and not duplicate the functionality in an application menu bar. In addition, the title of each application menu should be unique and not duplicate the titles that appear in the system menu bar. Finally, if the same menu appears in different menu bars within the application, the menu should have the same title and contain the same actions (worded the same and in the same order) wherever it appears.

7.1.1.3 Main Area

7.1.1.3.1 Arrangement of Window Controls

The main area of a window should be used to present text and/or graphic information and organized into subareas based on the nature of information being presented or the type of action required by users. Controls that perform similar or related functions should be arranged in sets and presented together in a control panel in a window, as shown in figure 7-2. The panel should be surrounded by a simple frame and clearly labeled to indicate the function performed by the controls presented in the panel. The controls should be grouped into one or more rows or columns as appropriate to the function performed by the controls.

Controls that refer to similar kinds of options should appear together in a set, and controls that refer to different kinds of options should be grouped separately. If scroll bars are needed, they shall be located to the right (for vertical scrolling) or at the bottom (for horizontal scrolling) of the area to be scrolled. If certain controls become unavailable, they should be dimmed and not available for selection. If certain controls are never available to users (e.g., if access to them is password controlled), they should not appear in the window.

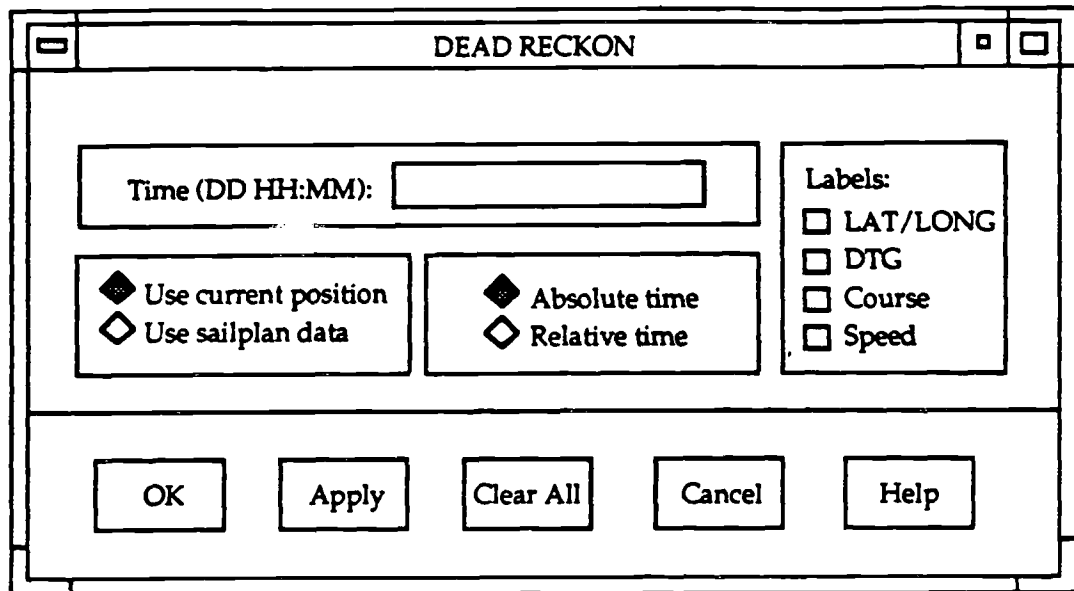


Figure 7-2. Example of grouping objects into control panels.

7.1.1.3.2 Arrangement of Push Buttons

The bottom of the main area should be reserved for push buttons indicating the actions that can be taken in the window. The push buttons should be displayed in a row that is centered at the bottom of the window and ordered from left to right based on the sequence in which they will be used, with the most frequently used button on the left. Buttons indicating positive actions should be toward the left, followed by buttons indicating negative actions and canceling actions. If a default push button is included in a window, it should be the leftmost button. A Help button should be included in every window and should appear at the right margin. The buttons should appear in the same order throughout the application.

In modal windows (i.e., those that interrupt the user's interaction with the application), push buttons should be arranged in the following order:

OK Cancel Help.

In modeless windows (i.e., those that do not interrupt interaction), the push button order should be:

OK Apply Cancel Help

or

OK Apply Reset Cancel Help.

The OK button should be the default in modal windows and in modeless windows

that are transient (i.e., displayed for only a single action). Apply should be the default in modeless windows that are likely to be displayed for multiple actions.

If users can perform multiple actions within a window and these actions can affect different objects or data elements in the window, push buttons should be labeled to reflect the object(s) that each push button affects. In addition, push buttons should be arranged in the window so that they are located near the object(s) to which they are related. Push buttons related to overall window functionality (e.g., OK, Cancel) should be arranged along the bottom of the window.

7.1.1.3.3 Arrangement of Action Icons

If action icons are used in an application, they should be bound to the application (i.e., not movable outside the application window in which they appear). An application should create no more than 20 action icon buttons, and they should appear along the left margin of the window, as shown in figure 7-3.

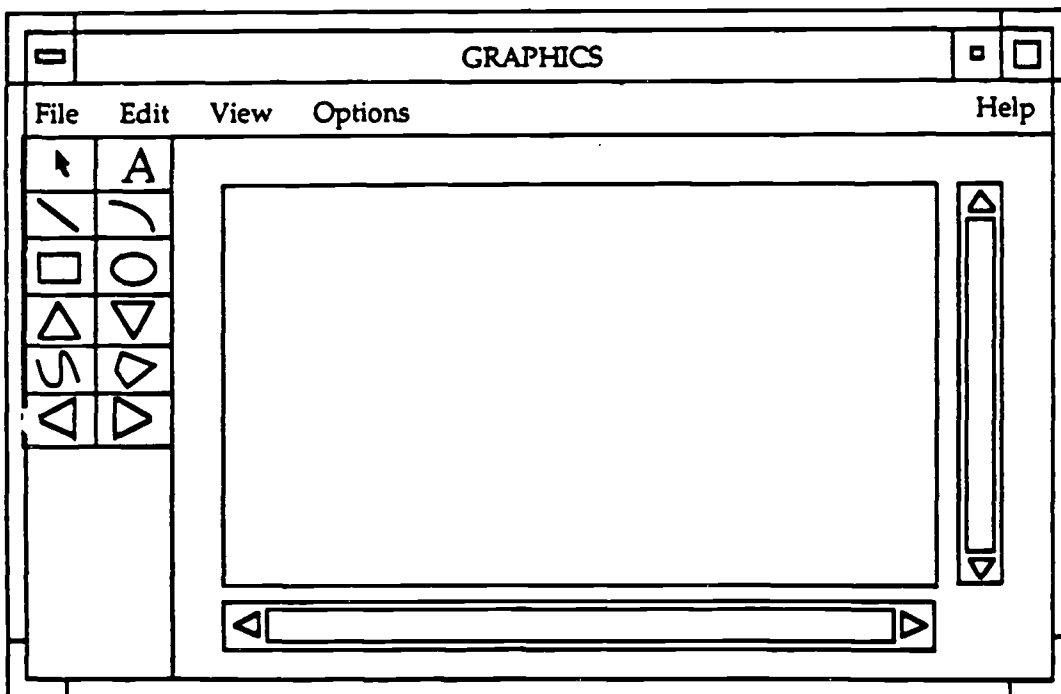


Figure 7-3. Recommended format for action icons in windows.

7.1.1.4 Message Bar

A message bar should be included along the bottom margin of a window to provide noncritical application messages to users. The left side of the bar should be used to present routine messages to users, provide simple help (e.g., actions needed to complete execution of a command), and indicate status pertaining to the window (e.g., feedback such as "Drawing map"). The right side of the bar should be used to identify information about the window (e.g., the name of the map or document, the page number such as "3 of 6"). A message bar should be part of the primary window(s) of an application but may also be included in a dialog window (as in figure 7-4). The message bar may be an area that is set aside in each window for this purpose, or may be an area that is used temporarily to present a message and then returned to its previous use.

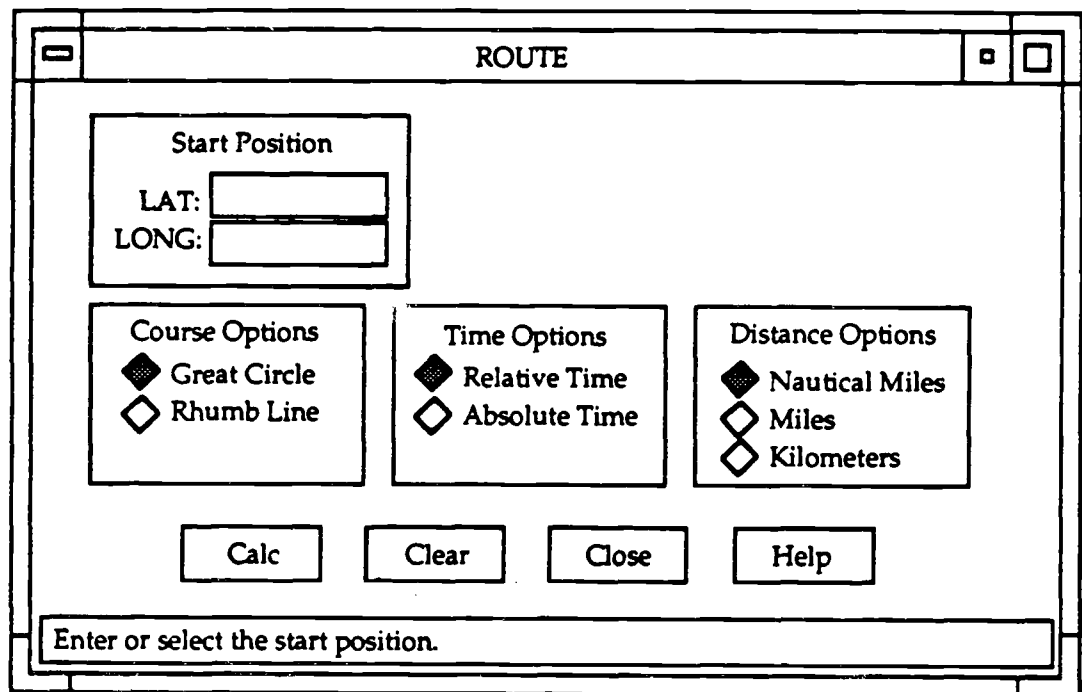


Figure 7-4. Example message bar in a dialog window.

7.1.2 Message Windows

When an application displays a message window, it should be as small as possible, yet large enough to display the information required. A message window should be uniquely identifiable by its appearance and be positioned at a standard location on the screen whenever it appears.

A message window should include a title, a symbol indicating the message type, the message itself, and one or more push buttons. The title of a message window should indicate the process or application that generated the message since multiple applications may be running simultaneously and users should be able to identify the source of the message (e.g., the message may be associated with an application that is iconified). The message itself should use language that is meaningful to users and require no further documentation or translation. Error messages should focus on the procedure for correcting the error, not the action that caused the error. If the error is repeated, the focus of error messages should change to include recommendations for preventive actions. A message window should include at least one push button, presented along the bottom of the window, that solicits a response from users.

7.1.2.1 Progress Message Windows

Applications should display a progress message window, shown in figure 7-5, when the processing time in response to a user's request exceeds five seconds. The window should contain a working symbol, a message, and the following push buttons (in the order indicated):

OK Cancel Help.

If the processing time will be lengthy, the window should be updated to indicate the status of processing (e.g., a message stating "5% sorted" or a scale showing the percent executed or transferred).

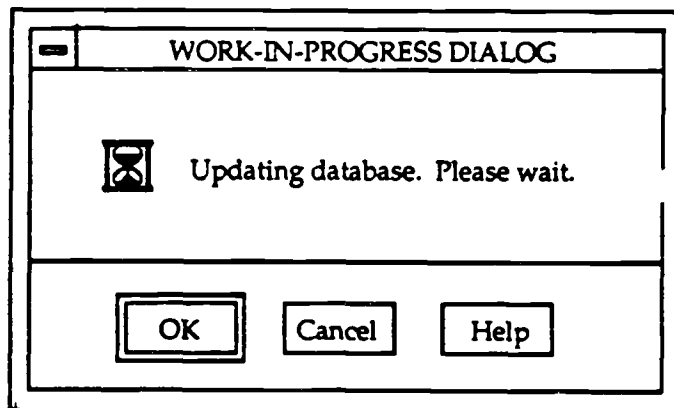


Figure 7-5. Example progress message window.

A progress message window should not interrupt processing by the application; the window should remain displayed until the action is complete or until the

window doing the processing is iconified. When processing is complete, the window should disappear. Users should be able to cancel the operation in progress if desired; they should confirm the action before it is executed if unsaved data will be lost.

7.1.2.2 Warning Message Windows

Applications should display a warning message window, shown in figure 7-6, to present critical messages concerning the consequences of an action. A warning window should be displayed when users attempt to execute a destructive action. The window should contain a warning symbol, a message, and the following push buttons (in the order indicated):

Yes No Help

or

OK Cancel Help.

When a warning window is displayed, processing by the application is usually suspended until users decide how to proceed.

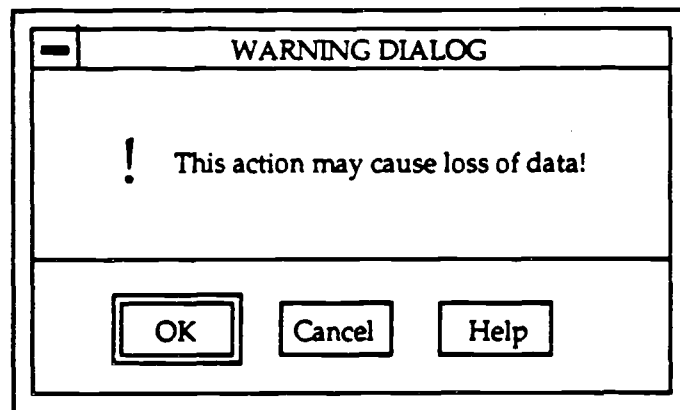


Figure 7-6. Example warning message window.

7.1.2.3 Error Message Windows

Applications should display an error message window, shown in figure 7-7, to inform users of an error that has occurred. The window should contain an error symbol, a message, and the following push buttons (in the order indicated):

OK Cancel Help

or

Retry Cancel Help.

The application should suspend processing until users respond to the window. If processing can be reinstated, the window should include a Retry push button.

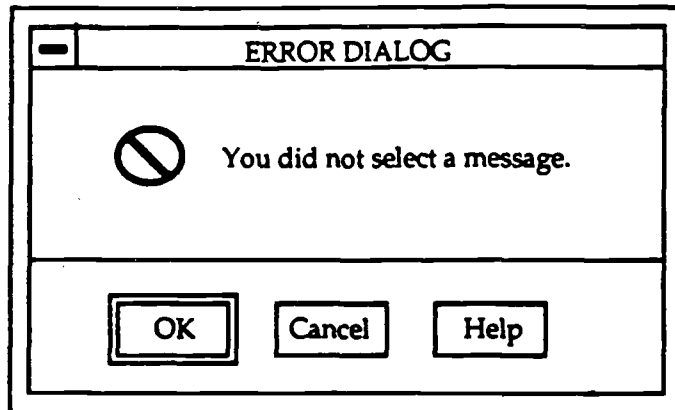


Figure 7-7. Example error message window.

7.1.2.4 Information Message Windows

Applications should display an information message window, shown in figure 7-8, to convey noncritical information that requires acknowledgement by users. (The message bar of a window should be used for frequently presented messages that require no acknowledgement by users.) The window should contain an information symbol, a message, and the following push buttons (in the order indicated):

OK Help.

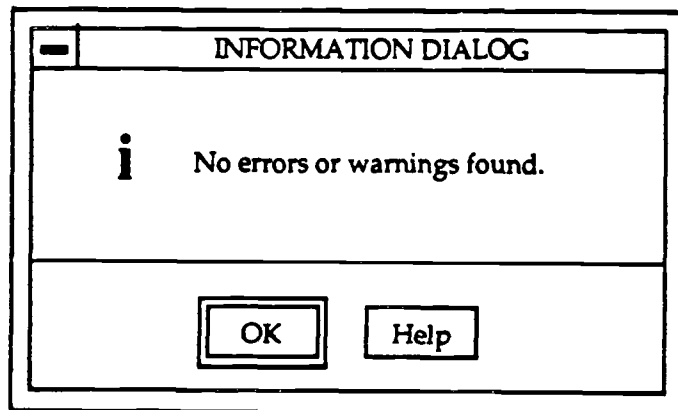


Figure 7-8. Example information message window.

An information message window should not interrupt processing by the application, and the window should remain displayed until users respond to it with

a specific action. Applications should not present timed-information windows (i.e., message windows that present information for a fixed time period) and then resume processing without requiring a user response.

7.1.2.5 Question Message Windows

Applications should display a question message window, shown in figure 7-9, to request clarification of a previous response. The window should contain a question symbol, a message, and the following push buttons (in the order indicated):

Yes No Help

or

Yes No Cancel Help.

The application should suspend processing until users respond to the window.

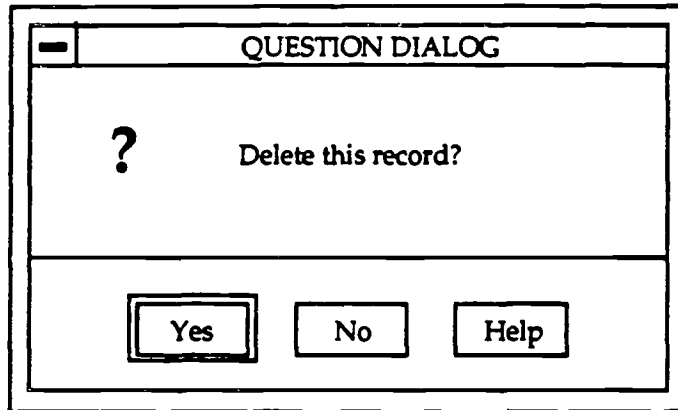


Figure 7-9. Example question message window.

7.2 APPEARANCE AND BEHAVIOR OF OBJECTS

7.2.1 Color of Objects

The interface objects (i.e., the window components, menus, and controls) in an application shall have the basic appearance indicated in table 7-1 and shown in figure 7-10.¹ Appendix A contains the Motif resource settings (and RGB values) that

¹The colors presented in these specifications are based on published guidelines concerning color use, and they represent an initial recommendation for implementation by ashore C² applications. These colors are being evaluated for their applicability to afloat C² systems (e.g., under different lighting conditions) and will be modified as needed.

Table 7-1. Appearance (color and 3-D effect) of objects in application windows.

<u>Object</u>	<u>Appearance and Behavior</u>
<u>Window Components</u>	
Title Bar	In an active window: DarkSlateBlue background, raised effect, LightGray text In an inactive window: LightBlue background, raised effect, LightGray text When selected: DarkSlateBlue background, depressed effect
Resize Border	In an active window: DarkSlateBlue background, raised effect In an inactive window: LightBlue background, raised effect When selected: No change in color or 3-D effect
Window Menu Button, Maximize Button, Minimize Button	In an active window: DarkSlateBlue background, raised effect In an inactive window: LightBlue background, raised effect When selected: DarkSlateBlue background, depressed effect
Window Menu	Basic appearance when displayed: LightBlue background, raised effect, Black text for available options, Gray text for unavailable options; first available option appears as a raised push button with LightBlue background, raised effect, Black text When an option is selected: Option becomes a push button with LightBlue background, raised effect, Black text
Client Area	Basic appearance: LightBlue background
Icon	Basic appearance: Icon image with LightBlue background, raised effect, Black text in the icon label
<u>Controls</u>	
Push Button	Basic appearance: LightBlue background, raised effect, Black text When selected: Gray background, depressed effect, Black text
Radio Button, Check Button	Basic appearance: LightBlue background, raised effect, Black text When selected: SlateGray background, depressed effect, Black text
Text Entry Field (Editable)	Basic appearance: LightGray background, depressed effect, Black text When text in the object is selected: Black background, LightGray text
Text Entry Field (Noneditable)	Basic appearance: Azure2 background, depressed effect, Black text When text in the object is selected: No change in text color
List Box	Basic appearance: LightGray background, depressed effect, Black text When an item is selected: Black background, LightGray text

Table 7-1. Appearance (color and 3-D effect) of objects in application windows (continued).

<u>Object</u>	<u>Appearance and Behavior</u>
Scroll Bar (see note below)	<p>Basic appearance: LightBlue background, raised effect for stepper arrow and slider; SlateGray background for scroll region</p> <p>When a stepper arrow is selected: SlateGray background, depressed effect</p> <p>When the slider is selected: No change in color or 3-D effect</p>
Scale	<p>Basic appearance: LightBlue background, raised effect for slider; SlateGray background, raised effect for scroll region; Black text</p> <p>When the slider is selected: No change in color or 3-D effect</p>
<u>Menus</u>	
Pull-Down Menu Bar	<p>Basic appearance: LightBlue background, raised effect, Black text</p> <p>When selected: Title appears as a push button with LightBlue background, raised effect, Black text</p>
Pull-Down Menu	<p>Basic appearance when displayed: LightBlue background, raised effect, Black text for available options, Gray text for unavailable options; first available option appears as a push button with LightBlue background, raised effect, Black text</p> <p>When an option is selected: Option becomes a push button with LightBlue background, raised effect, Black text</p>
Pop-Up Menu	<p>Basic appearance when displayed (with Right button): LightBlue background, raised effect, Black text for available options, Gray text for unavailable options; first available option appears as a raised push button with LightBlue background, raised effect, Black text</p> <p>When an option is selected (with Right button): Option becomes a push button with LightBlue background, raised effect, Black text</p>
Option Menu	<p>Basic appearance (of displayed option): Push button with LightBlue background, raised effect, Black text, bar graphic with LightBlue background, raised effect</p> <p>Basic appearance (when menu is displayed): LightBlue background, raised effect, Black text; option that was displayed is a push button with LightBlue background, raised effect, Black text</p> <p>When an option is selected: Option becomes a push button with LightBlue background, raised effect, Black text</p>

Note: The color of the stepper arrows and slider may be LightGray or Azure2 if the scroll bar is created from a convenience function (where these components share the color of their parent widget) rather than from a primitive widget (see Appendix A).

implement the appearance defined in the table. The shadow colors (i.e., for the top and left shadow and the bottom and right shadow) for each object shall be those that are generated automatically by Motif, and keyboard traversal shall not be part of an object's highlighted appearance. If necessary, applications may modify the basic object appearance defined here to provide the functionality desired; applications doing so should follow the guidelines provided in section 7.4 on color coding.

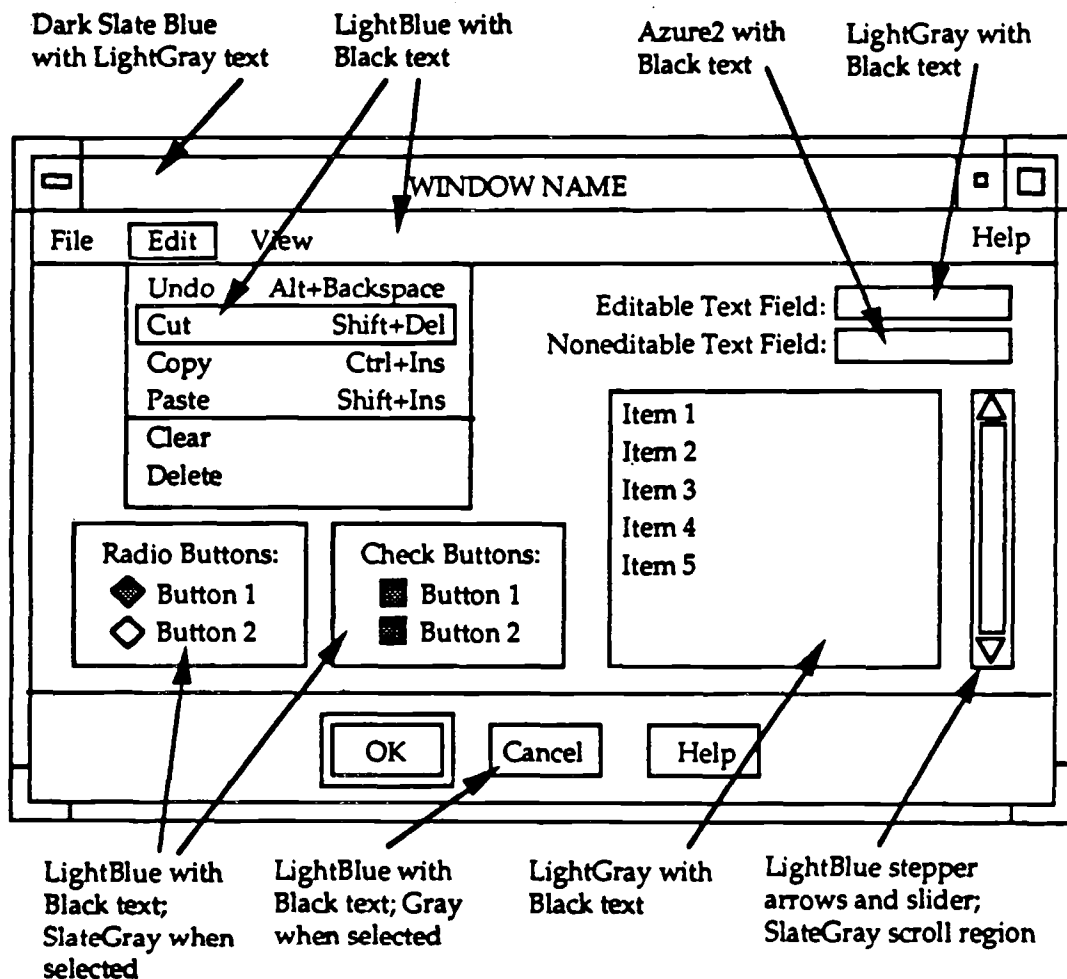


Figure 7-10. Example colors for interface objects.

7.2.2 Size of Objects

Menus should be wide enough to accommodate the wording of the longest menu option plus a keyboard accelerator, and should provide enough space between them so that each can be easily read by users.

Push buttons should be the same size within a window. If the push buttons contain text labels, the buttons should be wide enough to display the longest button label. If the push buttons contain icons, the buttons should be large enough to display the largest icon. Radio and check buttons should be the same size whenever they appear in a window.

Text entry fields and list boxes should be wide enough to read all of the text being presented. If necessary, horizontal and vertical scroll bars should be used to view the contents.

7.2.3 Consistent Appearance and Behavior

All objects in an application shall be identifiable solely on the basis of their appearance, and all objects of the same type shall have the same appearance. Users should be able to distinguish objects that are similar in shape (e.g., a push button and an option menu) on the basis of distinctive visual cues (e.g., an option menu should include a bar graphic). Users should not have to select an object in order to determine what it is or how it behaves.

Each application should determine the set of objects with which users will interact in performing a particular operation. These objects may vary between applications within a system. For example, one application may have users enter latitude and longitude by typing it in a text entry field while another application may have users enter these values by adjusting the slider bar in a scale. However, whenever an object is included in an application, it shall appear and behave in the manner indicated in these interface specifications. For example, selecting a menu title should always cause the menu to appear, and selecting a push button should always perform the action indicated in the button.

Application developers are free to create new objects to provide functionality not described in these specifications. However, developers shall not create a new object for functionality that already has an object associated with it, or use an existing object in a manner that conflicts with these specifications. When a new object is created, it should have an appearance and behavior that is consistent with that used for the basic set of objects presented in these specifications.

7.3 TEXT IN WINDOWS

7.3.1 Text Font, Size, and Readability

Normal, dimmed, and highlighted text shall appear and behave as indicated in table 7-1. In general, black text should be used in windows; if a different text color is used, it should have sufficient contrast with the background color to ensure good

readability.

The text in each window should be of sufficient size and thickness to be readable by users when they are seated at the normal viewing distance from the screen. Kobara in Visual Design with OSF/Motif recommends the use of a bold, sans serif font. The Army guidelines indicate that at a minimum, character height should be 1/200 of viewing distance; for example, a viewing distance of 36 inches requires a .18 inch character height. Text that is at least 14-point type should satisfy this requirement. In general, a bold font style (e.g., Times-Roman Bold) and large font styles improve readability. However, maximum text size should not exceed 10 percent of the available vertical display area of a full-size screen.

If the text is part of a window that is likely to be used in a briefing presentation, the text should be readable at the normal audience viewing distance when the text is projected on a large-screen display. Applications that cannot satisfy the range of viewing requirements with a single text font should provide text font size as a user-selectable option within the application.

7.3.2 Capitalization

Titles and major headings should be presented in upper-case letters; all other text (e.g., labels, directions, messages) should be presented in a combination of upper- and lower-case letters, following standard capitalization rules. A consistent approach to capitalization should be used so that the possibility of textual distraction for users is minimized. All upper-case letters should be used in text only for acronyms and abbreviations and for emphasis. Arabic rather than Roman numerals should be used when the information in a window has to be numbered.

7.3.3 Acronyms and Abbreviations

Acronyms and abbreviations should be used only when they are significantly shorter than the full word and they are commonly understood by users (e.g., are related to normal language or are specific job-related terminology). Abbreviations should be the shortest possible that will ensure uniqueness. Abbreviations should be meaningful and recognizable by users and should be used consistently within an application. Words not commonly abbreviated should not be abbreviated. Application developers should comply with the following documents for acronyms and abbreviations as appropriate to the application:

MIL-STD-12. Abbreviations for Use on Drawings, Specifications, Standards, and in Technical Documents

MIL-STD-411. Aircrew Station Signals

MIL-STD-783. Legends for Use in Aircrew Stations and on Airborne Equipment

New acronyms should be generated according to rules contained in MIL-STD-12. When abbreviations or acronyms are used, a dictionary should be available to users (e.g., in Help).

7.3.4 Noneditable Text

Applications should use consistent grammatical structure for all noneditable text (e.g., titles, labels, directions) displayed in windows. The wording should be consistent and use familiar terms and task-oriented language of users. Blocks of text should be broken into smaller, meaningful groups, as shown in figure 7-11.

Do this:	Not this:
<p>3 Bn Status</p> <ul style="list-style-type: none"> o Located at 32UNA100100 o Moving to Contact o 80% Strength o Supported by 9th armor platoon 	<p>The 3rd Bn is currently located at the 32UNA100100, moving to contact in sector 8, with 80% strength, supported by the 9th armor platoon.</p>

Figure 7-11. Example of breaking information into smaller text groups (from the Army guidelines).

If continuous text (in the form of complete sentences) is used in windows, it should be phrased in simple sentences, in the affirmative (rather than negative) and in active (rather than passive) voice, as shown in figure 7-12. A sequence of events or steps should be presented in the order they are performed. The referent for "it" or "they" in a sentence should be easily identified. Normal punctuation rules should be followed, and contractions and hyphenation should be avoided.

7.3.5 Editable Text

The manner in which editable text (i.e., text entered by users) is displayed in an application window should be appropriate to the task being performed in the window. For example, if users are doing normal word processing, the application should display the text exactly as it is typed by users (i.e., with the same

Do this:	Not this:
Press ENTER to continue.	The user should press ENTER to continue.
Clear screen before entering data.	Do not enter data before clearing the screen.
Select one.	Will you make a selection?

Figure 7-12. Example of wording style (from MIL-HDBK-761A).

capitalization and punctuation). However, if users are doing word processing to compose a Navy message, the application should display the text in capital letters (even if users type it in lower case) and with the appropriate punctuation. In addition, when presenting stored text in a window (e.g., in a text entry field), the application should display in a standard format (e.g., all capital letters), and any text editing performed by users should be converted into this format (e.g., text entered in lower-case letters should be converted into upper-case) by the application.

7.4 COLOR USE IN WINDOWS

Color (other than that specified in table 7-1) should be used in the main area of a window only to highlight, aid in grouping, clarify relationships, or add extra meaning. The arbitrary use of many colors may reduce the likelihood that significant color coding will be interpreted quickly and accurately by users. In general, applications should select colors to be consistent with the specifications contained in table 7-1. The following additional guidelines are provided on color selection and use:

- a. Color pairs at spectral extremes (e.g., red and green, yellow and purple) should not be used together because they appear to vibrate when placed next to each other.
- b. White text should not be displayed on a black background because this combination produces halation which makes the text less readable. In addition, because of focusing problems with blue, it should not be used as a text color or for any critical information.
- c. Color should be used consistently throughout an application and in ways that match user expectations. For example, red should be used to indicate critical or irreversible options; noncritical options should not be colored red

since users may be inhibited from selecting these options based on their color alone. In addition, the colors selected for status indicators should be the same throughout the application, and their use should be restricted to that function only.

- d. If color is used to distinguish message windows, it should appear in the message symbol (e.g., a warning hand in the warning message window) and not be used for message text. Red should be used only to alert users that an application or process is inoperative, possibly until appropriate corrective action is taken; yellow should be used only to alert users to situations where caution or recheck is necessary or unexpected delay will occur.
- e. If color is used to impart meaning, it should be used as a redundant code with another display feature (e.g., shape, symbology, content of text) and not as the sole basis for coding. When color coding is used, each color should represent one category of displayed data.
- f. The number of colors used to code the information in an alphanumeric display should not exceed seven, and only four codes should be displayed at any one time. The number of colors used to code information on graphical displays should not exceed eight or nine, since users have difficulty discriminating among more than this number of colors.
- g. Slight shade changes in color should not be used to show gradation or choice. Shade differences are usually difficult to see, especially on a varied background such as a map.
- h. The background color behind text should not be changed to show a change in system status because changing the background color usually reduces the readability of the text. Instead, the change should be signaled by changing the color of an object (e.g., a box or ball) next to the text.
- i. A black background should not be used when presenting color-coded items because black has poor contrast with many colors (e.g., red, blue). Background colors that provide high contrast improve readability. Unsaturated hues such as tan and offwhite are good choices as background colors; pure white should not be used because it is too bright.

7.5 CONSIDERATIONS IN WINDOW DESIGN

7.5.1 Selecting Objects to Match User Actions

Application developers should select the objects that appear in the window to match the actions that users will execute in the window. The following are guidelines on selecting objects to match user actions:

- a. Use primary window(s) for the primary actions within the application and to present controls that are used frequently. Use dialog windows for ancillary application actions and to present controls that are used less frequently.
- b. Use radio buttons when users need to choose one option from a small number (up to six) of mutually exclusive options; use an option menu when more options (no more than ten to twelve) are available. If space is limited in a window, use an option menu rather than radio buttons since an option menu takes very little space. Use a list box when users need to make a choice from a large group of options (more than twelve).
- c. Use radio buttons or an option menu when the set of options from which to choose is not likely to change. Use a list box when the options may change.
- d. Use check buttons when users need to choose more than one item from a small number of options (up to seven) in a group; use a list box otherwise.
- e. Use push buttons in a control panel for frequently used selections or when users are already moving the pointing device in the control panel area. Use pop-up menus only when it is critical to the application that users be able to access functions without moving the pointing device. Do not use pop-up menus for frequently executed operations because they require users to recall where the functionality is provided and how to access it.
- f. Use option menus for setting values or choosing from a set of related items and not for selecting commands.

7.5.2 Arranging Information to Match User Actions

Application developers should determine how users will execute the actions called for by a window and then design the window layout so that users can manipulate objects in ways that support task performance. For example, if an application generates information that should be presented a page at a time, the developer should provide users with the ability to page by including push buttons that perform standard paging operations (e.g., Next, Previous).

The window layout should be designed according to users' natural scanning order and probable selection sequences. In most cases, this order should be from left to right and top to bottom. For example, when presenting a series of radio or check buttons, the most frequently used button should be the top (or leftmost) button in the group. The window layout should be designed from the perspective of what is logical to users and appropriate to the actions being executed, not what is logical or appealing to application developers.

7.5.3 Arranging Information by Importance

The most important information and controls associated with the task being performed in a window should be located in the upper left part of the window, working down to the less important information at the bottom of the window. In addition, the objects in a window should be arranged to accommodate the possibility that users will resize the window. Placing the most important objects in the upper-left corner of the window allows users to manipulate these objects even if the window is reduced to a size that is smaller than normal. Finally, if a window contains task-critical information, the information should be visually set apart from other information in the window so that it can be easily seen by users. The minimum separation should be one character space above and below and two character spaces before and after the critical information.

7.5.4 Designing Windows to Minimize User Memory Load

Windows should be designed to minimize the short-term memory load placed on users when they perform the task called for by the window. The window should contain all of the information relevant to the task and should allow the user to complete the task without having to refer to information not included in the window.

Figure 7-13 presents an example of a window that places an unnecessary memory load on users. This window allows users to select the colors to be assigned to countries appearing on a map display. Users select a set of countries, choose a color, and then execute the selection on the map; the selections are cleared from the window, and users repeat this process for the next color. The window design and selection process require users to know the country abbreviations and to remember both the countries they have already selected and the colors they have assigned to these countries. In addition, users have to check the map display to ensure they do not inadvertently select a country a second time and assign an incorrect color to it. A better design for this window would be to present a matrix of countries and colors so that users can make all of their selections at once. This design would minimize the memory load placed on users and reduce the likelihood of user error because all

of the information would be available in the window and users would be able to review their selections as they made them.

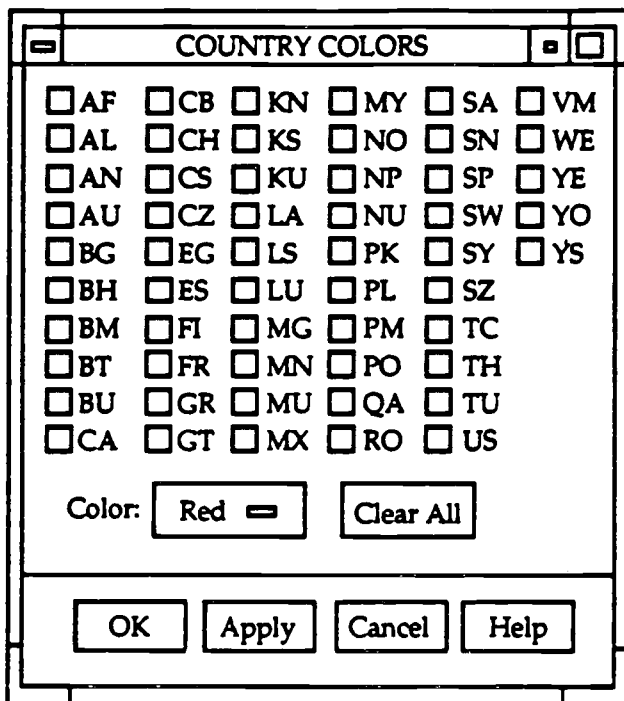


Figure 7-13. Map countries window.

7.5.5 Coding Critical Information in Windows

Capitalization should not be used as the sole indication of critical information in a window. Attention should be focused on critical information by bolding/brightening, color coding, or other means. Both color (e.g., red) and sound (e.g., audio alarms) should be used for critical messages, and users should be required to respond before an audio alarm is terminated.

Applications shall use auditory signals to alert users to critical conditions or operations. If auditory signals are used for noncritical operations (e.g., as an alternate means of information presentation), users should be provided with a simple means to acknowledge and turn off the signal at their discretion. Auditory signals should be intermittent in nature and allow sufficient time to respond; they should be distinctive in intensity and pitch, and the number of signals provided to users should not exceed four. The intensity, duration, and source location of the

signal should be selected to be compatible with the acoustical environment of the users and the requirements of other personnel in the area surrounding the system.

Flash coding should be used only to display urgent information for user attention. No more than two levels of coding should be used. MIL-HDBK-761A indicates that the flash rate should be in the range of 3-5 Hz with equal on and off intervals; if two flash coding levels are used, the second should flash at less than 2 Hz. IEEE recommendations indicate the flash rate should be within 1-5 Hz with a minimum cycle of 50 percent. When flash coding is used on a displayed item, a flashing marker symbol (such as asterisks) should be used rather than blinking the information itself. Users should be able to acknowledge the event causing the flashing and suppress it if desired.

7.5.6 Dynamic Information in Windows

When a window contains dynamically changing information, an application should allow users to control the rate of update based on the use to be made of the information. In addition, an application should allow users to freeze the display of any information that is being updated automatically and to resume the updating either at the point of stoppage or at the current point in time.

MIL-HDBK-761A recommends that when dynamically changing information must be reliably and accurately read, the values that users read should not be updated more often than once per second. If users have to identify the rate of change or read gross values, the values should be updated at a rate of between two and five times per second.

Applications should be able to alert users to critical information that becomes available in an inactive or iconified window in the application. If a dynamic window is temporarily frozen (e.g., while executing a print command), the application should provide immediate feedback to users and should prompt them to return to automatic updating. Users should be informed if significant changes in data occurred while the display was frozen.

7.5.7 Consistency in Design Across Windows

Application developers should adopt a consistent organizational scheme for the key elements in a window and then apply that basic scheme to all windows in the application. The same window design should be employed whenever users have to perform the same basic task; for example, a single design should be used to present identifying information on data records, whether the data are tracks or messages. Different or distinctive elements can appear in a window to fit the task being

performed, but these elements should be consistent across windows within the application.

7.6 WINDOW MANAGEMENT CONSIDERATIONS

7.6.1 Initial Window Appearance

Application developers should determine the appropriate appearance for each application window when it first appears on the screen. Developers should consider what information should be visible. For example, when a scrollable window containing text appears on the screen, the window should display the topmost part of the information and should be wide enough for users to read a complete line of text without scrolling.

In addition, developers should determine the appropriate settings for the objects in a window when it appears on the screen. For example, when radio or check buttons are included in a window, developers should determine if the application should save the button settings in their most recently executed state or return them to a default (or unselected) state when the window next appears on the screen. However, when users select button(s) but do not execute the selection (e.g., select Cancel), the application should not save the selection and the button(s) should revert to their initial state when the window was displayed.

7.6.2 Initial Window Size and Placement

When a window is displayed, it may fill all or part of the available screen in the system window. An application window that covers all of the screen should contain frame borders so that users can resize it and access any other windows that it might cover.

Whenever possible, each window in an application should be sized so that all of the objects in the window are visible when the window first appears and users can see the whole page with which they are working. At a minimum, a window should be wide enough for users to read the title, and should be tall enough for users to read the information in the title bar and menu bar.

Application developers should determine the initial placement for each window when it first appears on the screen.² In deciding on a window's placement, developers should consider the importance of the information presented in the

² Application developers determine the placement of each window the first time it appears on the screen. Thereafter, a window is displayed at the same location where it was when it was last closed or iconified.

window (e.g., important information should be placed at the center of users' visual focus), the other information already displayed (e.g., important information should not be obscured), and the distance from the current pointer location (e.g., the amount of pointer movement required by users to execute an operation should be minimized).

7.6.3 Resizing

When users change the size a window, only the location of the window borders should change, not the size or relative position of the objects within the borders. For example, if a text window is enlarged, more lines of text should be visible in the window, but the size of the text should not change. The contents of the window should remain visible during resizing so that users can view the effect of the change in size on the amount of information that can be seen. Horizontal and vertical scroll bars should appear as appropriate when a window is resized to be smaller than the information being presented in the window. In addition, the extent to which a window can be resized should be limited so that objects are not obscured from view when the window is at its minimum size. In windows that have multiple panes, application developers should determine how the space in the window is divided between panes after users resize the window. Depending on the functionality (e.g., mapping), an application may require the size and/or contents of the panes to change automatically based on the size of the window.

7.6.4 Order of Window Interaction

Application developers should determine for each dialog window in the application the set of other windows with which users are allowed to interact while the dialog window is displayed. Developers should specify if interaction with other windows is unrestricted (i.e., modeless) or if interaction is not allowed with any parent window (i.e., primary modal), any window in the application (i.e., application modal), or any window in the system (i.e., system modal).

7.6.5 Processing in Iconified Windows

Applications should be designed so that users are kept informed when processing is ongoing in an iconified window. Users should be informed whenever critical processing events (e.g., when processing stops, when an error occurs, when additional input is needed) occur in an iconified window.

8.0 TASK-SPECIFIC WINDOWS

8.1 HELP WINDOWS

8.1.1 System-Level and Application-Level Help

Help should be available to users at both the system level and within individual applications. Help should be provided in the following ways:

- a. As a menu title in the system menu bar. This level of help should describe system capabilities and provide information on how to use help, and may include an on-line tutorial for users and a system navigation aid (such as the one shown in figure 6-5) that serves as a summary reference from which users can select information on different system functions.
- b. As a menu title in an application menu bar. This level of help should include general information on application functionality and may include an online index with cross-referenced listings so that users can obtain information on a particular window, action, or command. If desired, applications using action icons can provide access to help via an action icon rather than a menu title.
- c. As a push button in a dialog window. This level of help should provide information on the actions that can be taken in the window.
- d. As a message in the message bar of a window. This level of help should explain how to complete the execution of an action such as a menu command (e.g., "Click on the map, then drag the pointer over the area to be zoomed.").

8.1.2 Window Design

A help window, such as that shown in figure 8-1, should include a window title that identifies the contents, a main area that is scrollable if the amount of information exceeds the size of the window, and an OK push button at the bottom of the window to remove it from the screen. The push button should appear and behave as the default action so that users can close a help window by pressing the Return key.

When a help window appears on the screen, it should be wide enough for users to read the text in the window; it should display the information at the beginning of the Help description and be placed on the screen so that the window does not obscure the object(s) it is describing. Users should be able to move and resize a help window. In addition, users should be able to keep the help window on the screen

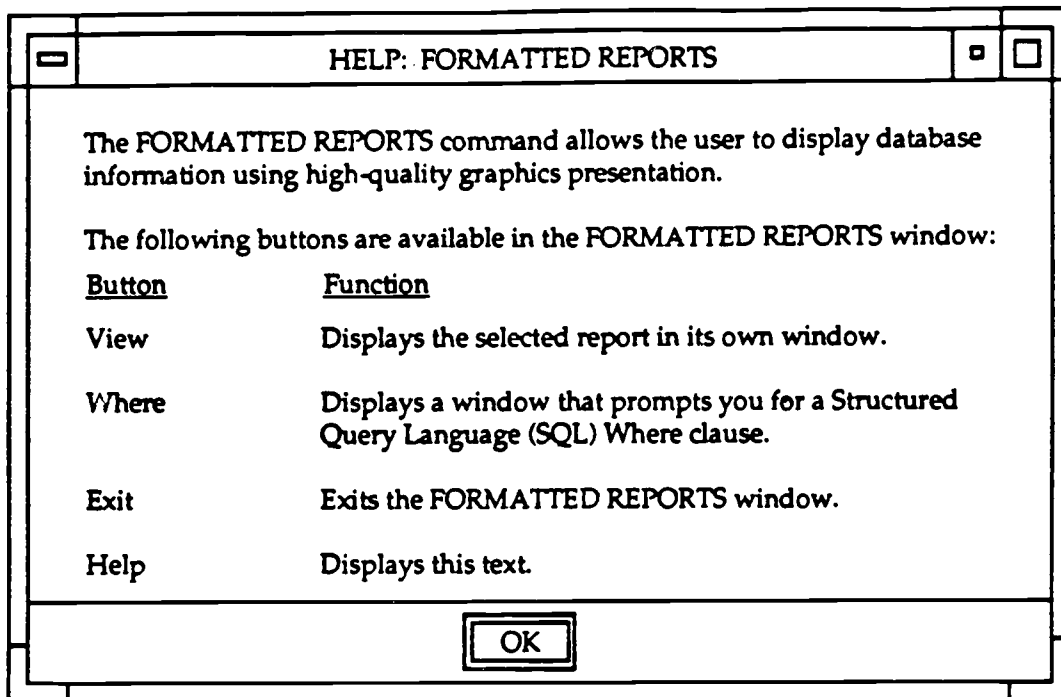


Figure 8-1. Example help window.

while working in an application window (e.g., so they can follow the sequence of steps listed in help). Users should be able to print the content of any help window, either by selecting all of the text in the window or by marking the beginning and end of the portion of interest. When done with help, users should have an easy means to return to the window(s) in which they were working when they requested help.

8.1.3 Window Content

The help window should explain the steps required to execute the action(s) in the window about which users requested help. The information presented should be limited to the actions that can be performed in the window. The information should be worded concisely, using consistent grammatical structure, phrasing, and punctuation. For example, figure 8-2 provides an example of inconcise wording; figure 8-3 presents a more concise version of the same information. When presenting a sequence of steps, the explanation should follow the same sequence (e.g., save data before quitting; or save data, then quit the application).

The information presented in a help window should assume that users have received previous training and need to be reminded of the proper information or procedure. Users should be able to easily scan to the text of interest (e.g., capitalize it

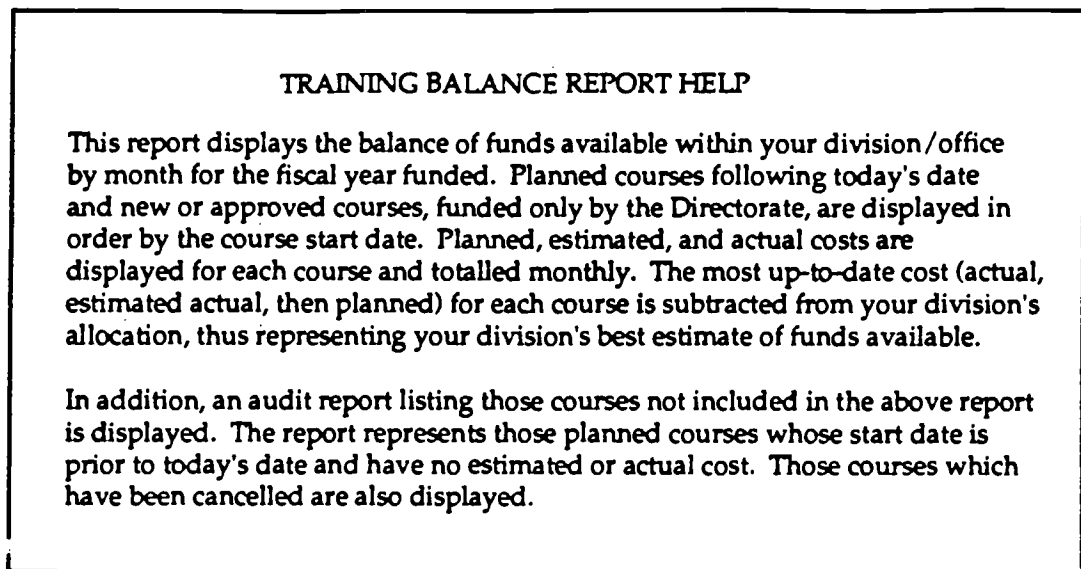


Figure 8-2. Example of inconcise wording of help information (from MIL-HDBK-761A).

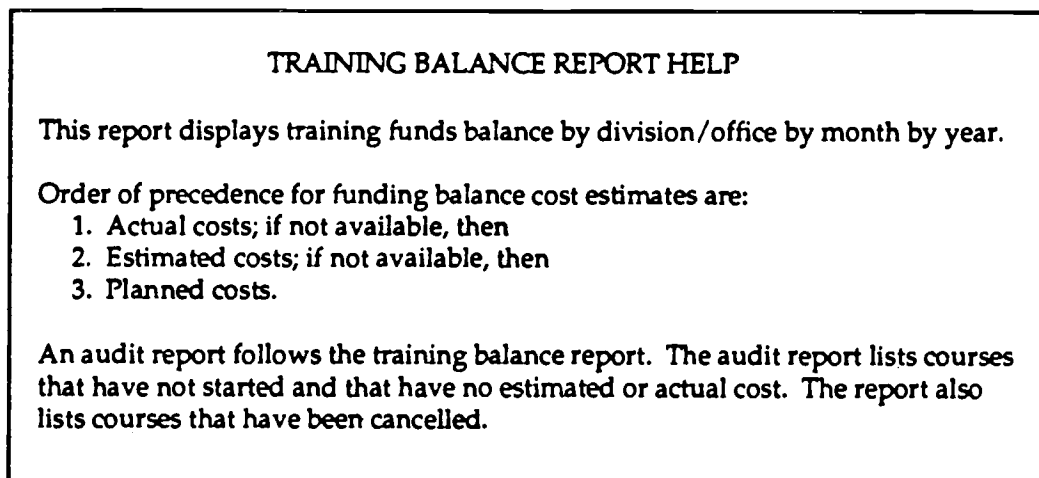


Figure 8-3. Example of concise wording of help information (from MIL-HDBK-761A).

to make it easier to find), and the steps should be numbered and presented as separate paragraphs if more than two steps are involved. Help should decode all abbreviations and acronyms used in the window about which users requested help, and should explain how each user-selectable object in the window behaves. A help window should be removed from the screen when the window about which users requested help is iconified or closed.

8.1.4 Online Training

Online training should be available at both the system and application levels and should be provided as a module that is separate from help. Online training should be designed to accommodate different levels of user expertise and should adapt to user ability. It should use simulated data and protect the real data in the system from change or corruption. The training mode should be clearly indicated to users and kept separate from normal system use.

8.2 DATA ENTRY WINDOWS

8.2.1 Window Design

A data entry window, such as that shown in figure 8-4, consists of a template that is used to display, enter, change, and delete data. The template consists of a set of data fields, each consisting of a field name and an area where the data value appears. The data fields may be text entry fields, option menus, or information-only areas, depending on the type of data being presented.

REPORTED POSITION					
Procedure	Action	Record	Page	Field	Help
CURRENT POSITION/HISTORY					
Track:	<input type="text"/>	Name:	<input type="text"/>		
Type:	<input type="text"/>	Hull:	<input type="text"/>		
UIC:	<input type="text"/>	Class:	<input type="text"/>		
DTG	LAT	LONG	Sensor	Course	Speed
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Figure 8-4. Example data entry window.

A data entry window should include a title that describes the contents of the window. If the data fields extend beyond a single page (i.e., what can be displayed in the window without scrolling), each page should have the same title and be numbered to indicate the current page and total number of pages (e.g., 1 of 3). If possible, related data fields should appear on the same page. The numbering of items should be continuous from one page to the next (e.g., if the last numbered item on page 1 is #5, the first numbered item on page 2 should be #6).

The design of each data entry window should be consistent with the task being performed by users. Data fields should be organized by sequence of use, frequency of use, or importance, with related fields appearing together and separated from unrelated fields. If users are entering data from a hardcopy form, the window format should be identical to the hardcopy format, as shown in figure 8-5.

Hardcopy Form at:

DATABASE RECORD	
Data Field 1	
Data Field 2	
Data Field 3	
Data Field 4	Data Field 5

Window Format:

DATABASE RECORD	
Data Field 1:	<input type="text"/>
Data Field 2:	<input type="text"/>
Data Field 3:	<input type="text"/>
Data Field 4:	<input type="text"/> Data Field 5: <input type="text"/>

Figure 8-5. Consistency in layout of hardcopy form and data entry window (from the Army guidelines).

If data must be entered in tabular form, the entry areas should be arranged in rows and columns, with each row and column labeled. If the data fields extend beyond a single page, the row and column labels should remain along the edges of the window and should not disappear from view when the window is scrolled. Every fifth row of data should be separated by a blank line or other delimiter. The information in each field should be automatically justified, and numeric data should be automatically justified according to the decimal point.

A data entry window should contain the controls (such as push buttons or pull-down menus) needed to support data entry and manipulation. For example, if a data entry window takes more than one page, the window should include controls that provide for paging. In addition, if users have to make multiple entries, the data entry window should include controls to clear the window and restart data entry. If an application uses push buttons as the controls in a data entry window, they should appear along the bottom of the window and be visually separated from data entry areas.

A data entry window should be designed so that users can obtain information about a data field and its contents. Options available for providing this information are to automatically display it in the message bar of the window when the text insertion cursor is placed in a field or to have users click on the data field name, select a Help push button in the window, or select Help from an application-level menu or action icon.

8.2.2 Data Field Format

Each data field name used in a data entry window should be unique, and the same name should be applied to the same field throughout the application. Data fields should be identified as mandatory or optional, and optional fields should be labeled as such. Data fields with values that can be changed should appear as text entry fields or option menus, depending on whether users have to type the value or can select it from a limited set of values. Data fields with values that cannot be changed should appear as information-only areas.

8.2.3 Data Entry and Manipulation

Users should execute the same set of actions to enter and manipulate data in data entry windows throughout the application. Users should be able to place the text insertion cursor using the pointing device and move the cursor using either the keyboard or the pointing device. Users should perform text entry in text entry fields as indicated in section 5.4.2, move between text entry fields as indicated in sections 2.1.2.3 and 5.4.3, and make selections from option menus as indicated in section 4.3.

Users should be able to tab to adjacent fields, and tabbing should move through the fields in the same direction (e.g., left to right, top to bottom) in all data entry windows in the application. Users should be able to back up to any field and change its contents prior to entering the data into the system. The application should perform a validity check on the data entered (e.g., numeric data meet range [minimum-maximum] requirements, abbreviations and acronyms have been reviewed by an online spellchecker) and provide a visual and/or auditory error message when an entry is invalid.

When users finish making entries in the appropriate data fields in the window, they should enter the data into the system with an explicit action such as selecting a Save menu option or clicking on an Apply or OK push button. If appropriate, the application should provide feedback to users (e.g., a confirmation message) to indicate successful data entry. Users should be able to enter data into the system at any time during data entry.

8.2.4 Data Query

If an application provides a data query capability, the language provided to users should reflect the structure of the data as perceived by users. The language should allow users to specify the data to retrieve from system storage, then to display and manipulate (e.g., edit, delete) it as appropriate. Users should be able to indicate what data are requested without having to tell the system how to find it. In addition, they should be able to construct queries using terminology that is operationally meaningful rather than using terms or formats that reflect how the data are stored by the system.

Users should be able to construct simple and complex queries that retrieve the desired information, create predefined queries, and save, retrieve, and execute these queries. The language should permit users to construct alternate forms of the same query that correspond to the variations available in natural language. The application should prompt users to confirm that a query should be processed if the time required for data retrieval will be excessive.

8.3 TEXT WINDOWS

8.3.1 Window Design

When a text document is displayed in a window (e.g., in a word processing application), the window should be wide enough to display an entire line of text without having to scroll the window horizontally. The window should provide scroll bars for scrolling vertically through the document, and a standard location

(e.g., the message bar of the window) where users can determine their current location within the document (e.g., current page number).

Users should be able to save, access, retrieve, and rename text documents as well as be able to print the document. The user should be able to specify print options (e.g., font), the portions of the document to be printed (e.g., pages 2-9 of a 10-page document), and the printer where the document will be sent (if more than one printer is available). The system should display an acknowledgement of the print command and provide users with the ability to check the status of the printer (online/offline) and the printer queue.

8.3.2 Text Manipulation

It is expected that, where appropriate, systems will provide the full range of text manipulation capabilities (e.g., in a word processor) through the use of COTS software. When an application allows users to perform text manipulation, it should provide the basic capabilities described in this section.

Users should be able to easily specify the format of a document (e.g., margins, tabs) and to select the font type, size, and style (e.g., bold, italics) for text. Automatic line break and wrap-around when the text being entered reaches the right margin as well as automatic pagination, with page numbers based on the number entered by users, should be available. The text that is displayed in the window should either be formatted to reflect how the document will appear when it is printed (i.e., what you see is what you get), or users should have the option to display the text in this format before deciding to print the document. A copy of the original document should be retained until users confirm that it is to be changed; the document should not be modified automatically as users make each editing change.

Applications should provide both search and search/replace capabilities for users. In the former case, users type the text string to be searched, and the application locates and highlights the first instance of the string in the text. The search should not be case sensitive. In the latter case, users type both the text string to be searched and the text string to which it is to be changed. The search should be case sensitive; e.g., users should be able to change "navy" to "Navy."

8.4 TABULAR DATA WINDOWS

8.4.1 Window Design

A tabular data window, such as that shown in figure 8-6, should be used to organize and display alphanumeric information in tabular or columnar form. This type of window should be used for information display only; a single- or multi-

columned list box (see section 5.5) should be provided if users are expected to select one or more of the items.

Part Name	Part Number	Bin Number	Units
Power supply, 28 VDC	M34 564	234	5
Signal amplifier	G34 672	814	8
Signal converter	344 W34	332	7
Transmitter cable	W88 909	448	1
12-pin connector	RS4 445 2	129	1
Connector adapter	MTO 344	786	0
Switching unit	R34 556 7	550	5
Antenna assembly	344 56 334	878	8
Antenna mounts	342 998	134	3

2 of 10

OK Next Previous Cancel Help

Figure 8-6. Example tabular display window (from MIL-HDBK-761A).

The window should include a title describing its contents, and vertical and horizontal scroll bars if the information being presented exceeds the space available in the window. If the data in a window extend vertically across several pages, each page should be numbered, and the window should include push buttons for paging commands. In addition, the columns should be labeled identically on each page, and the last line of one page should be the first line on the succeeding page. Whenever possible, the content of the window should be arranged so that it does not extend over more than one page horizontally. If the data being displayed are the output to a database query, the window should include the query that was executed.

8.4.2 Window Content

Each column in a tabular data window should have a heading, and the data in one column should be clearly separated from that in other columns (usually by at least three spaces). Data groupings should be indicated with blank space, separator lines, and/or different intensity levels. Multiple colors should be used only if the colors provide additional meaning. If the window contains many rows and

columns, a blank space should be inserted after every third to fifth row and column to facilitate scanning by users.

Alphabetic data should be left-justified within a column, numeric data without decimals shall be right-justified, and numeric data with decimals shall be justified by the decimal point, as shown in figure 8-7. Long strings of numbers should be delimited with spaces, commas, or slashes to facilitate readability, and leading zeros should not be used unless required for clarity. If the data in a record extend beyond a single line, each line beyond the first one should be indented or in some way identified as a continuation of the record.

Do this:	Not this:
Artillery Tanks Jeeps Aircraft	Artillery Tanks Jeeps Aircraft
400 4210 38 3911	400 4210 38 3911
1.5 10.38 1.365 .500.0	1.5 10.38 1.365 500.0

Figure 8-7. Example of data justification within columns (from the Army guidelines).

The data in a tabular display window should be grouped or arranged so that users can identify similarities, differences, trends, or relationships. For example, depending on the purpose of the window, data may be presented in sequential, spatial, alphabetical, functional, or chronological order. Data that are particularly important, require immediate user response, and/or are used more frequently should be grouped at the top of the window. If the data extend across several pages, related data should be placed on the same page. If an application allows users to display the same information as they entered (e.g., in a data entry window), the format for the display should be compatible with the format used for data entry. The

ordering and layout of the corresponding fields should be consistent, and the names of individual fields should be the same.

8.5 GRAPHIC DISPLAY WINDOWS

8.5.1 Window Design

A graphic display window should be used when presenting information in the form of a line graph, bar chart, histogram, or flow chart. The window should include a title describing its contents and should be sized whenever possible so that the entire graphic display is visible when the window appears on the screen; users should not have to scroll or resize a graphic display window to view its contents.

8.5.1.1 Line Graphs

Applications should use line graphs, such as those shown in figure 8-8, for presenting trend information, spatially structured information, time critical information, or relatively imprecise information. The axes of the graph should be clearly labeled and include the unit of measurement as appropriate. The labels should be in upper- and lower-case letters and oriented left to right (including the vertical axis of a graph) for normal reading. The minimum and maximum value should be indicated on each axis, with up to nine intermediate markings showing gradations on the axis. The starting point on each axis should be zero, with the gradations indicated in whole numbers, unless a zero starting point is inappropriate for the data being displayed. The gradations should be at standard intervals (e.g., 1, 2, 5, 10), with intervening gradations consistent with the labeled scale interval.

Labels should be used instead of legends or keys when it is necessary to identify the data that are plotted on the graph. The labels should be oriented horizontally and located next to the data being referenced. Each line or curve on a graph should be labeled and coded (e.g., with solid, dashed, dotted lines), and a line or curve containing critical or abnormal data should be coded (e.g., by color, line thickness, annotation) to call attention to that part of the display. If grid lines are included in a graph, they should be unobtrusive and not obscure the data presented in the graph; users should be able to display or suppress grid lines as desired. The same coding scheme should be used consistently within an application whenever the same types of data are presented graphically.

If users are required to compare multiple trend lines, the lines should be presented on a single graph. If the lines have to be presented in multiple graphs, users should be able to redraw the graphs using the same scale on both graphs to facilitate comparison. If users have to read precise values from a graph, users

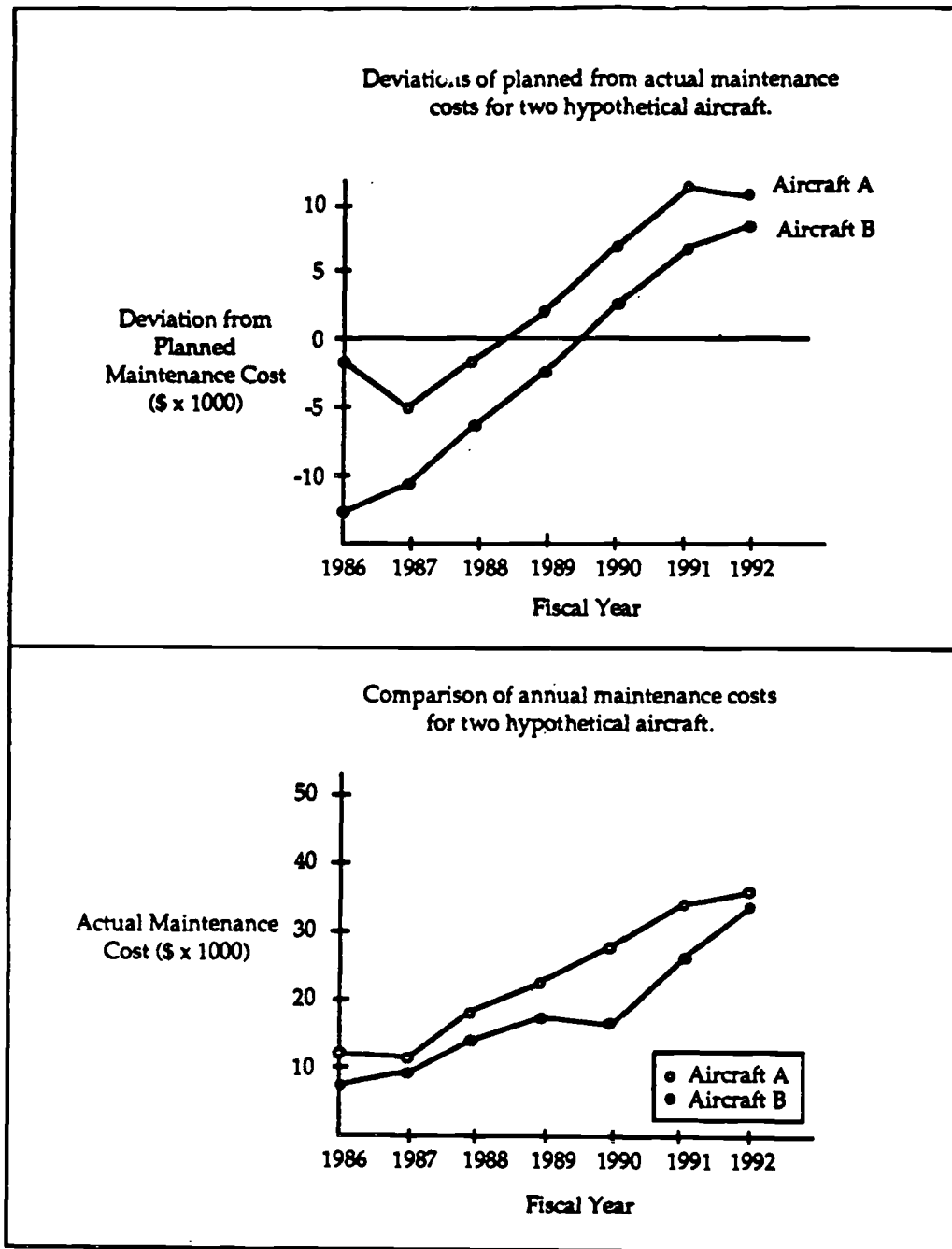


Figure 8-8. Example line graphs (from MIL-HDBK-761A).

should have the option to display the actual data values on the graph and to zoom the display if necessary.

8.5.1.2 Bar Charts and Histograms

Applications should use a bar chart, such as the ones shown in figure 8-9, for comparing a single measure at several intervals, and a histogram for comparing a single measure when the number of intervals is large. The bars in a related set of bar graphs should have a consistent orientation (vertical or horizontal) and bars containing data that must be compared should be presented adjacent to one another. If the displayed data have to be compared with a critical value, a reference index should be provided.

The bar chart or histogram should be designed to conform to user expectations. For example, the use of icons (e.g., a silhouette of a person to represent 1,000 people) to represent quantitative information should be avoided. Charts and axes should be clearly labeled, and important information should be highlighted. When bars are presented in pairs, they should be labeled as a unit, with a legend provided that distinguishes between the bars.

8.5.1.3 Flow Charts

Applications should use a flow chart for presenting a schematic representation of sequences or processes. The steps in a flow chart should be presented in a logical order. If there is no inherent logic, the steps should be ordered to minimize the size of the flow chart. The path indicated in the flow chart should be left to right, top to bottom, or clockwise. Each decision point in the flow chart should contain a single, simple decision, as shown in figure 8-10. The elements and lines should be coded (e.g., symbol and shape coding) to assist in understanding, and the same coding scheme should be used throughout the flow chart. For example, the flow chart should provide directional indicators (arrows) to indicate the sequence to be followed. A legend should be included that describes each element and code used in the flow chart, and critical information and/or steps should be highlighted.

8.5.2 Manipulation of Graphical Data

It is expected that where appropriate, systems will provide the full range of graphic manipulation capabilities through the use of COTS software. When an application allows users to manipulate graphic data, it should provide the basic capabilities described in this section.

Users should be able to select and edit the attributes of objects (e.g., color, line thickness, font size), change object sizes (enlarge or reduce), and fill the enclosed

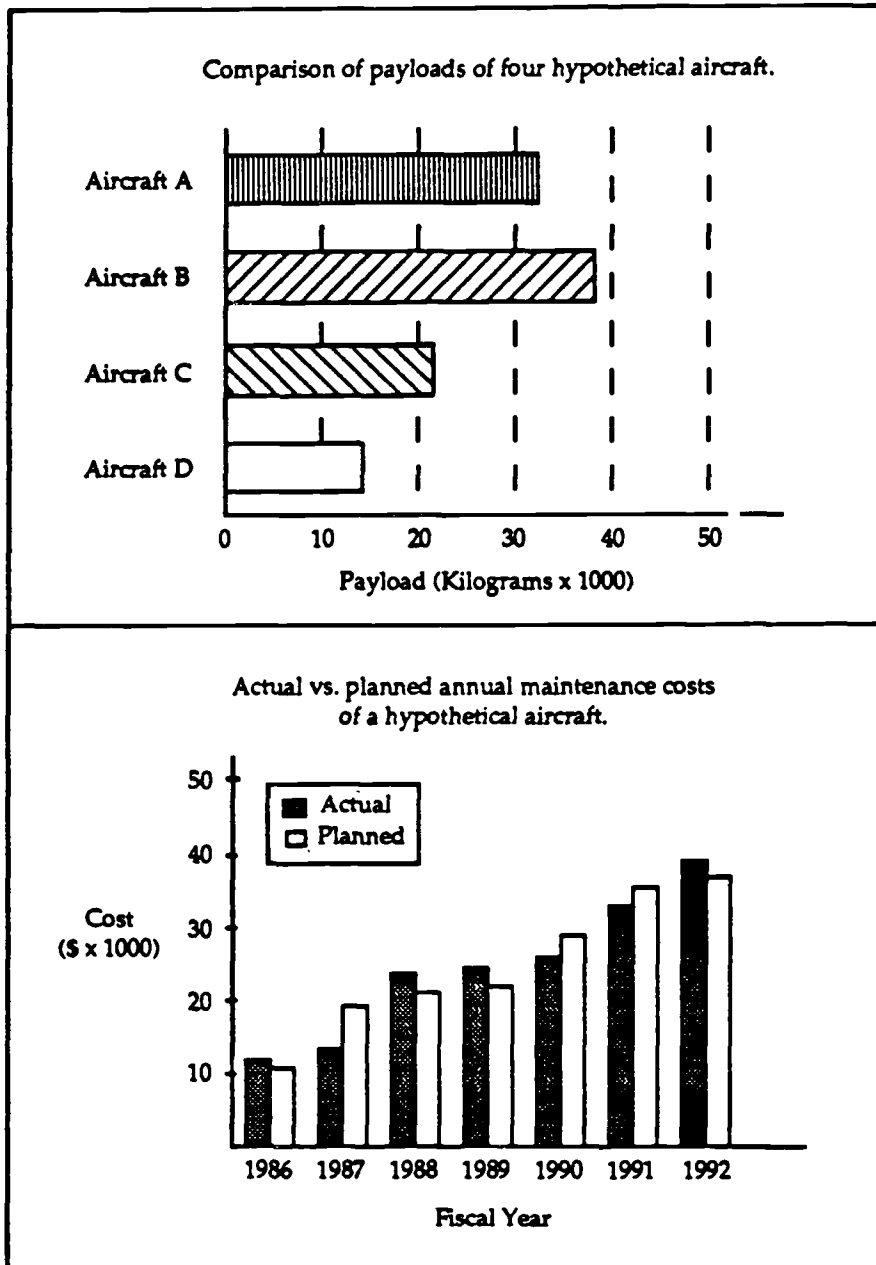


Figure 8-9. Example bar charts (from MIL-HDBK-761A).

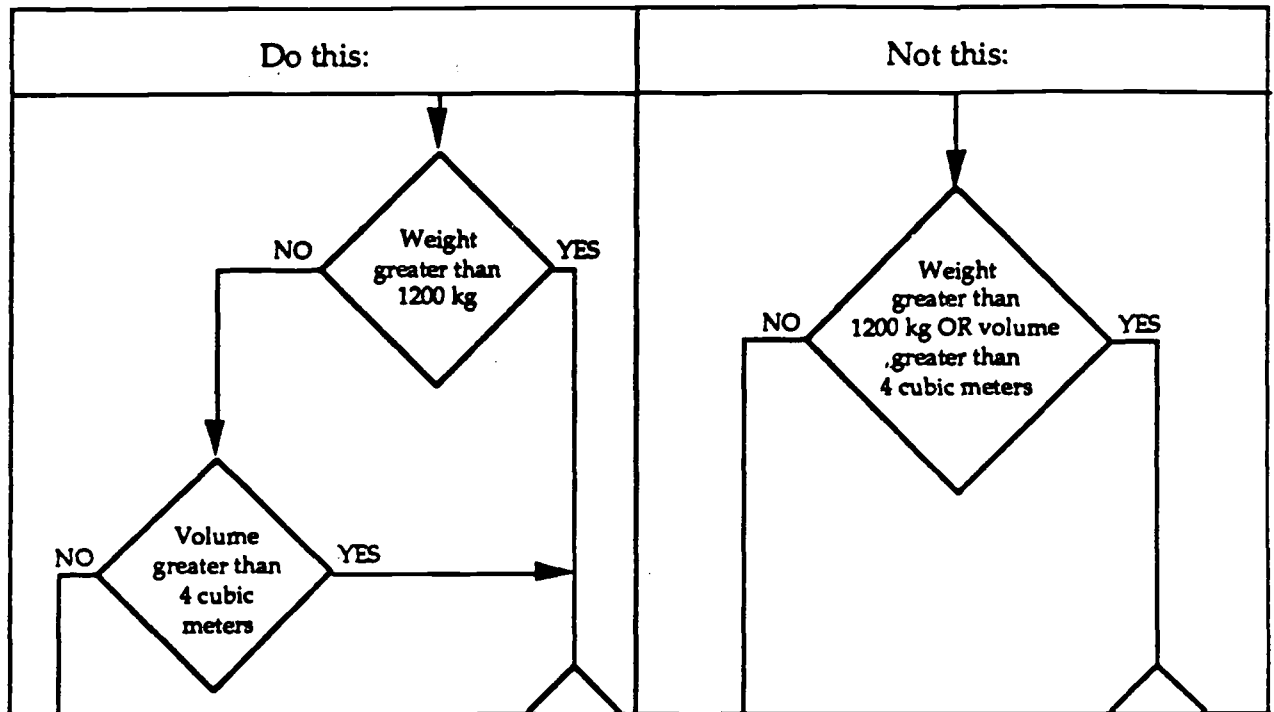


Figure 8-10. Example of labeling decision points in a flow chart (from MIL-HDBK-761A).

area with colors or patterns. Applications should automatically align objects to an invisible rule or grid structure, complete figures (e.g., closure of a polygon), and draw lines between user-specified points.

Users should be able to draw objects such as lines, rectangles, ovals, and arcs; these objects should emerge as they are drawn and be easy to reposition, duplicate, and delete. Users should be able to group separate objects into a single object (e.g., combine text and a rectangle into a complex object). The objects that are displayed in the window should either be formatted to reflect how they will appear when printed (i.e., what you see is what you get), or users should be provided with an option to display the objects in this format before deciding to print them. A copy of the original graphics should be retained until users confirm that the objects are to be changed; the objects should not be modified automatically as users change them.

8.6 MAP WINDOWS

8.6.1 Window Design

A map window may contain both the map display and a set of controls for manipulating the map, or the entire window may be devoted solely to the map, with map controls available in individual dialog windows. A map window should include identifying information about the map (e.g., map name, coordinates, area, scale) along with status information (e.g., "drawing map"). This information may be presented in a subarea within the window (as in figure 8-11) or may be included along the bottom margin of the window. A continuous coordinate indicator giving the pointer location on the map should be available in a standard area of the window (e.g., with the identifying information at the bottom of the map).

Each map should be displayed using the same orientation (i.e., north toward the top of the window), and the important features should be labeled. The labels should

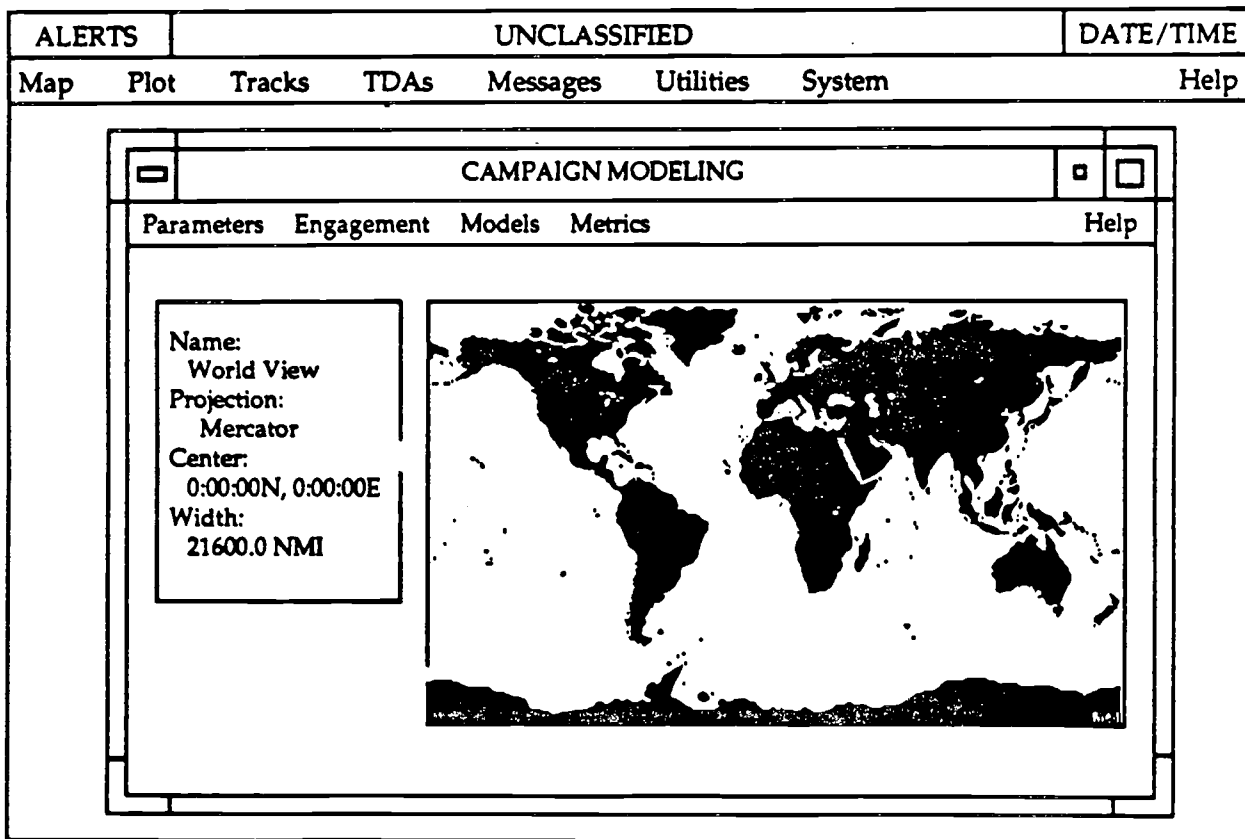


Figure 8-11. Example map window with identifying information.

be positioned consistently with respect to the feature they describe (e.g., next to or below the feature), should not obscure important information or clutter the map, and should remain legible at all map resolutions.

When users display a color overlay on a map, a color coding key should also be displayed (e.g., in a subarea of the map window or in a separate dialog window) so that users can interpret the information in the overlay. Users should be able to display the coding key as desired without having to redisplay the overlay. If the coding key is presented in a dialog window, the window should be the minimum size needed to present the required information and positioned so that it obscures as little of the information on the map as possible. If appropriate, the key should function as a scale so that users can interpret the coding in the overlay easily and accurately.

Standard symbology (e.g., NTDS, NATO) and color codes (e.g., red for hostile contacts) should be employed, with help available to users for identifying unknown symbols or other map information. If multiple symbol sets are available, users should be able to select the set desired and to switch between sets without losing data. Symbols should be placed on the map accurately or connected to the desired location using arrows, lines, or other graphics. Symbol labels should appear next to the symbol and present essential information (e.g., unit or track identification) about the symbol. The background of the symbol and label should be transparent so that it does not obscure other information (such as overlays) displayed on the map.

8.6.2 Map Manipulation

Users should be able to customize a map window to fit the task being performed. They should be able to pan and zoom a map as desired. A position or change indicator should be provided as a means for users to return quickly to the normal or starting map. When a map is zoomed, the size of symbols, labels, and other map features should be adjusted so that users can read them. In addition, users should be able to define a baseline (i.e., home) position on a map and be able to return to this position quickly (e.g., by selecting the appropriate menu item). Users should be able to place the window where desired, define the appearance of the map (e.g., select country colors), select the objects that will appear on the map (e.g., tactical overlays, contact data), and change the appearance of critical tactical information on the map (e.g., the color of friendly/hostile/neutral contacts) as appropriate to the task being performed.

Users should also be able to add, edit, reposition, and delete labels and overlays on the map and determine the distance and bearing between any two points on the map. Latitude and longitude should be enterable to the level of accuracy needed (e.g., degrees, minutes, seconds, tenths or hundredths of seconds), and when

calculations such as range, bearing, and position are performed, the answer that is computed should reflect the degree of accuracy appropriate to the scale of the map displayed.

Users should be able to select and deselect symbols on the map by following the methods described in table 2-2. Users should be able to view or declutter overlapping symbol labels and obtain additional information, including exact map coordinates, for selected symbols on the map (e.g., by double clicking on the symbol). The intensity of the map should be adjustable so that selected portions of the map can be faded out without losing all map features. Users should be able to distinguish among symbols that represent coincident points and to obtain information that will allow them to resolve ambiguities among the symbols.

8.6.3 Automatic Updating

Users should be able to select the categories of tactical information that will be automatically updated and specify the frequency and rate at which the information is updated. If appropriate to the application, users should be able to temporarily stop and then resume the updating process.

8.7 MESSAGE HANDLING WINDOWS

8.7.1 Message Preparation

Windows supporting the preparation of military messages, such as the one shown in figure 8-12, should follow the same general design as data entry windows, and users should be able to enter message information using the same data entry procedures.

Users should be provided with a basic set of message header fields and should be supported whenever possible in specifying the message address. For example, option menus should be available for selecting from limited sets of frequently used terms (e.g., classification level), and when replying to a message, the appropriate addressee(s) should be provided automatically. Users should be able to build and maintain distribution lists of commonly used addresses and select from these lists (without having to reenter the information) when addressing messages. The address should be checked for accuracy prior to transmission, and users should have to correct any errors before the message is sent.

Preformatted forms that conform to standard message format should be available; when users enter the text of a message, format control should be automatic. Users should be able to specify the data to be transmitted in the message,

The screenshot shows a window titled "MESSAGE ENTRY FORM" with a menu bar containing "Procedure", "Action", "Record", "Page", "Field", and "Help". The main area is divided into two sections:

JANAP 128 FORMAT LINE 1:

- Channel Designator: 24C
- Channel Sequence Number: 087

JANAP 128 FORMAT LINE 2:

- Action Precedence: 1
- Language Media Format: US
- Classification: U
- Content Indicator Code/ Communications Action Identifier: 23CF
- Originator Station Routing Indicator: LCRPLCL

A vertical scrollbar is on the right side of the main area, and a "3 of 12" indicator is at the bottom right.

Figure 8-12. Example message preparation window.

incorporate existing file data (including other messages received or transmitted) if desired, and save the message during preparation and upon completion.

8.7.2 Message Transmission

Message transmission procedures should be designed to minimize the user actions required. Users should be able to initiate message transmission directly (e.g., by selecting a Transmit push button). If a message cannot be sent immediately, it should be queued automatically so that users do not have to actively monitor the transmission and undelivered messages will be saved in the event of transmission failure. Users should be able to assign message priorities and to cancel or abort a transmission that has not been completed. Feedback should be available on status of message transmission, confirming that messages have been sent and indicating when transmission failures occur. Users should be able to specify what feedback they want to receive, and an automatic log of this information should be maintained.

8.7.3 Message Receipt

Users should be informed when high priority messages are received. For example, during login, a system may provide users with a list of new messages received since they last accessed the system. In addition, during a session, a system may activate an alert indicator in the system window to inform users of priority messages. Message notification should not interfere with ongoing system use but should provide some indication of urgency if the messages have different degrees of priority.

Incoming messages should be automatically queued by time of receipt and message priority, and logs of this information should be maintained. Users should be able to review summary information on messages that have been queued, display individual messages, save/file the ones of interest, and discard those that are unwanted. When a message is displayed, it should appear in a text window, and users should be able to scroll, save, and print it as they would any other text document.

APPENDIX A MOTIF RESOURCE SETTINGS

This appendix provides the Motif resource settings that implement the “look and feel” called for by the user interface specifications presented in this document. The settings listed in table A-1 define the appearance and behavior of the Motif window manager, including the appearance of the system window as described in table 6-1. The settings listed in table A-2 define the appearance (primarily the color and 3-D effect) of the Motif primitive widgets from which the objects listed in table 7-1 are constructed. In some cases, the settings indicated in the tables are the Motif default for the resource. Developers shall build resource files to comply with the resource configuration presented in this appendix. If developers need to create new interface objects or modify existing ones to provide the functionality desired, the appearance of these objects should be consistent with the resource settings provided in this appendix.

The primary goals in developing the color scheme presented here were to select colors that make objects easy to distinguish from their background and that provide an indication of an object’s functionality (e.g., objects that users can select are assigned the same color) while, at the same time, conforming to Motif design philosophy and utilizing Motif default settings whenever possible. However, efforts to satisfy the first two criteria were unsuccessful because they resulted in objects that lacked consistency in appearance.

Motif provides the capability to build objects using either widgets or gadgets. Widgets have flexibility in the set of colors that can be assigned to an object but do not provide the programming efficiency that can be achieved through the use of gadgets. However, gadgets are limited by Motif to share the colors of their parent widgets. As a result, when an attempt was made to color code objects by functionality, the same object sometimes had different colors depending on whether it had been created using a widget or a gadget. Efforts to use color coding in the manner desired were discarded, and a color scheme was selected that would ensure consistency in object appearance regardless how the object was created.

It was expected that commercial products such as word processing and graphics software would be integrated with applications that comply with the specifications presented in this document. These products are likely to have a user interface that conforms with Motif guidelines concerning color use. If the color scheme defined here is too different from that normally used by Motif, the user interface will lack a consistent appearance when software from different sources is integrated into a single system. Although a color scheme could have been selected that improved the discriminability of objects, this effort produced objects that did not conform to Motif guidelines and would have resulted in a user interface with an inconsistent

appearance. As a result, the color scheme that was selected made only minor modifications to Motif design guidelines.

The settings in tables A-1 and A-2 were derived from information in the OSF/Motif Programmer's Reference, release 1.1. In addition, recommendations provided by Kobara in Visual Design with OSF/Motif are included in the two tables where they are consistent with the specifications contained in this document. Developers should consult this reference for guidance on how to build a user interface that is consistent with Motif design philosophy.

The following comments describe the key features of the resource configuration provided in tables A-1 and A-2:

(1) The color scheme presented here utilizes the following colors (and their RGB values) for interface objects: LightBlue (173 216 230), DarkSlateBlue (72 61 139), Azure2 (224 238 238), LightGray (211 211 211), Gray (192 192 192), SlateGray (112 128 144), and Black (0 0 0). In addition, the color scheme calls for shadow colors based on the Motif color algorithm for the colors that are used.

(2) Color coding was used only to distinguish between objects with similar shapes but different functions (e.g., editable text has a LightGray background and noneditable text has an Azure2 background). In addition, where possible, colors were selected to improve contrast between objects (e.g., Black text on a pale background) and to focus attention on particular objects (e.g., a DarkSlateBlue frame to emphasize the active window).

(3) Keyboard traversal (which allows the user to move between objects when using the keyboard) is set to false for all objects except editable text because the specifications provided in this document do not require keyboard-only interaction. However, highlight thickness is set to 2 pixels to assist in consistent sizing and placement of widgets, as recommended by Kobara.

(4) If specific colors are entered for the top and bottom shadow colors (rather than using the Motif default which sets them dynamically), the colors selected should match those generated when the Motif color algorithm is used.

Table A-1. Resources and settings for the Motif window manager.

<u>Resource</u>	<u>Setting</u>
<u>Appearance of Window Manager Components</u>	
Mwm*background	LightBlue
Mwm*bottomShadowColor	Default (Dynamic)
Mwm*foreground	LightGray
Mwm*topShadowColor	Default (Dynamic)
Mwm*activeBackground	DarkSlateBlue
Mwm*activeBottomShadowColor	Default (Dynamic)
Mwm*activeForeground	LightGray
Mwm*activeTopShadowColor	Default (Dynamic)
Mwm*client*background	LightBlue
Mwm*client*foreground	Black
Mwm*feedback*background	LightBlue
Mwm*feedback*foreground	Black
Mwm*menu*background	LightBlue
Mwm*menu*foreground	Black
<u>Window Manager Behavior</u>	
Mwm*autoKeyFocus	False
Mwm*buttonBindings	Default ("DefaultButtonBindings")
Mwm*cleanText	Default (True)
Mwm*clientAutoPlace	False
Mwm*colormapFocusPolicy	Default (Keyboard)
Mwm*deiconifyKeyFocus	Default (True)
Mwm*enforceKeyFocus	Default (True)
Mwm*fadeNormalIcon	True
Mwm*focusAutoRaise	Default (True)
Mwm*frameBorderWidth	6 (per Kobara)
Mwm*iconAutoPlace	Default (True)
Mwm*iconClick	Default (True)
Mwm*interactivePlacement	Default (False)
Mwm*keyBindings	Default ("DefaultKeyBindings")
Mwm*keyboardFocusPolicy	Default (explicit)
Mwm*limitResize	Default (True)
Mwm*lowerOnIconify	Default (True)
Mwm*moveThreshold	Default (4)
Mwm*multiScreen	True
Mwm*passButtons	True
Mwm*passSelectButton	Default (True)
Mwm*positionIsFrame	Default (True)
Mwm*positionOnScreen	Default (True)
Mwm*raiseKeyFocus	Default (False)

Table A-1. Resources and settings for the Motif window manager (continued).

<u>Resource</u>	<u>Setting</u>
Mwm*resizeBorderWidth	7 (per Kobara)
Mwm*resizeCursors	Default (True)
Mwm*showFeedback	Default (all)
Mwm*startupKeyFocus	Default (True)
Mwm*traversalOn	False
Mwm*useClientIcon	Default (False)
Mwm*useIconBox	Default (False)
Mwm>windowMenu	Default ("DefaultWindowMenu")
Mwm*wMenuButtonClick	Default (True)
Mwm*wMenuButtonClick2	Default (True)
<u>Icon Appearance</u>	
Mwm*iconDecoration	Default (label image)
Mwm*iconImageMaximum	Default (50x50)
Mwm*iconImageMinimum	Default (16x16)
Mwm*iconPlacement	Default (left/bottom)
Mwm*icon*background	LightBlue
Mwm*icon*foreground	Black

Table A-2. Resources and settings for Motif widgets.

<u>Resource</u>	<u>Setting</u>
<u>Generic Settings</u>	
*background	LightBlue
*cursor	left_ptr
*fontList	9x15bold, 8x13bold recommended
*foreground	Black
*pointerColor	Black
<u>ArrowButton</u>	
Resources inherited from XmPrimitive:	
XmNbottomShadowColor	Default (Dynamic)
XmNforeground	LightBlue
XmNhighlightThickness	Default (2)
XmNshadowThickness	Default (2)
XmNtopShadowColor	Default (Dynamic)
XmNtraversalOn	False
Resources inherited from Core:	
XmNbackground	SlateGray
XmNborderWidth	Default (0)
XmNheight	16 recommended (per Kobara)
XmNwidth	16 recommended (per Kobara)
<u>BulletinBoard</u>	
New resources:	
XmNallowOverlap	False recommended (per Kobara)
XmNmarginHeight	2 recommended (per Kobara)
XmNmarginWidth	2 recommended (per Kobara)
Resources inherited from XmManager:	
XmNbottomShadowColor	Default (Dynamic)
XmNshadowThickness	Default (1 if the parent is a DialogShell; 0 otherwise)
XmNtopShadowColor	Default (Dynamic)
XmNtraversalOn	False
Resources inherited from Core:	
XmNbackground	LightBlue
XmNborderWidth	Default (0)

Table A-2. Resources and settings for Motif widgets (continued).

<u>Resource</u>	<u>Setting</u>
<u>CascadeButton (in a Menubar)</u>	
Resources inherited from XmLabel:	
XmNalignment	Default (XmALIGNMENT_CENTER)
XmNlabelType	Default (XmSTRING) recommended (per Kobara)
XmNmarginHeight	Default (2) recommended (per Kobara)
XmNmarginWidth	8 recommended (per Kobara)
XmNmnemonic	Default (NULL)
Resources inherited from XmPrimitive:	
XmNbottomShadowColor	Default (Dynamic)
XmNforeground	Black
XmNshadowThickness	Default (2)
XmNtopShadowColor	Default (Dynamic)
XmNtraversalOn	False
Resources inherited from Core:	
XmNbackground	LightBlue
XmNborderWidth	Default (0)
<u>CascadeButton (in a Menupane or Popup Menu)</u>	
New resources:	
XmNmappingDelay	Default (180)
Resources inherited from XmLabel:	
XmNalignment	Default (XmALIGNMENT_CENTER)
XmNlabelType	Default (XmSTRING) recommended (per Kobara)
XmNmarginHeight	Default (2) recommended (per Kobara)
XmNmarginWidth	Default (Dynamic) recommended (per Kobara)
XmNmnemonic	Default (NULL)
Resources inherited from XmPrimitive:	
XmNbottomShadowColor	Default (Dynamic)
XmNforeground	Black
XmNshadowThickness	Default (2)
XmNtopShadowColor	Default (Dynamic)
XmNtraversalOn	False
Resources inherited from Core:	
XmNbackground	LightBlue
XmNborderWidth	Default (0)

Table A-2. Resources and settings for Motif widgets (continued).

<u>Resource</u>	<u>Setting</u>
<u>DrawnButton</u>	
Resources inherited from XmLabel:	
XmNalignment	Default (XmALIGNMENT_CENTER)
XmNmarginHeight	Default (2)
XmNmarginWidth	Default (2)
XmNmnemonic	Default (NULL)
Resources inherited from XmPrimitive:	
XmNbottomShadowColor	Default (Dynamic)
XmNforeground	Black
XmNshadowThickness	Default (2) recommended (per Kobara)
XmNtopShadowColor	Default (Dynamic)
XmNtraversalOn	False
Resources inherited from Core:	
XmNbackground	LightBlue
XmNborderWidth	Default (0)
<u>Form</u>	
Resources inherited from XmManager:	
XmNbottomShadowColor	Default (Dynamic)
XmNshadowThickness	Default (Dynamic)
XmNtopShadowColor	Default (Dynamic)
XmNtraversalOn	False
Resources inherited from Core:	
XmNbackground	LightBlue
XmNborderWidth	Default (0)
<u>Frame</u>	
Resources inherited from XmManager:	
XmNbottomShadowColor	Default (Dynamic)
XmNshadowThickness	2 recommended for In and Out Frames, 1 for Etched In and Etched Out Frames (per Kobara)
XmNtopShadowColor	Default (Dynamic)
XmNtraversalOn	False
Resources inherited from Core:	
XmNbackground	LightBlue
XmNborderWidth	Default (0)

Table A-2. Resources and settings for Motif widgets (continued).

<u>Resource</u>	<u>Setting</u>
<u>Label</u>	
New resources:	
XmNalignment	XmALIGNMENT_BEGINNING for vertically stacked labels and labels with widgets to their left; XmALIGNMENT_END for labels with widgets to their right; otherwise XmALIGNMENT_CENTER recommended (per Kobara)
XmNlabelType	Default (XmSTRING) recommended (per Kobara)
XmNmarginBottom	Default (0)
XmNmarginHeight	6 recommended (per Kobara)
XmNmarginWidth	Default (2) recommended (per Kobara)
XmNmnemonic	Default (NULL)
Resources inherited from XmPrimitive:	
XmNforeground	Black
XmNshadowThickness	Default (0)
XmNtraversalOn	Default (False)
Resources inherited from Core:	
XmNbackground	LightBlue
XmNborderWidth	Default (0)
<u>List</u>	
New resources:	
XmNlistMarginHeight	2 recommended (per Kobara)
XmNlistMarginWidth	2 recommended (per Kobara)
XmNlistSizePolicy	XmCONSTANT recommended (per Kobara)
XmNlistSpacing	Default (0)
XmNscrollBarDisplayPolicy	XmSTATIC recommended (per Kobara)
Resources inherited from Primitive:	
XmNbottomShadowColor	Default (Dynamic)
XmNforeground	Black
XmNhighlightThickness	Default (2)
XmNshadowThickness	Default (2)
XmNtopShadowColor	Default (Dynamic)
XmNtraversalOn	False
Resources inherited from Core:	
XmNbackground	LightGray
XmNborderWidth	Default (0)

Table A-2. Resources and settings for Motif widgets (continued).

<u>Resource</u>	<u>Setting</u>
<u>PanedWindow</u>	
New resources:	
XmNmarginHeight	Default (3)
XmNmarginWidth	Default (3)
XmNsashHeight	6 recommended (per Kobara)
XmNsashIndent	Should equal the value by which vertical Scrollbars are offset from the right edge of the Paned Window; -4 if no Scrollbars are used (per Kobara)
XmNsashShadowThickness	Default (Dynamic)
XmNsashWidth	Should equal the width of the Scrollbars in the application; 12 if no Scrollbars used (per Kobara)
XmNseparatorOn	Default (True)
XmNspacing	Default (8) recommended (per Kobara)
Resources inherited from XmManager:	
XmNbottomShadowColor	Default (Dynamic)
XmNshadowThickness	Default (2) recommended (per Kobara)
XmNtopShadowColor	Default (Dynamic)
XmNtraversalOn	False
Resources inherited from Core:	
XmNbackground	LightBlue
XmNborderWidth	Default (0)
<u>PushButton</u>	
New resources:	
XmNarmColor	Gray
XmNdefaultButtonShadowThickness	Default (0)
XmNfillOnArm	Default (True)
XmNshowAsDefault	Default (0)
Resources inherited from XmLabel:	
XmNalignment	Default (XmALIGNMENT_CENTER)
XmNmarginHeight	Default (2) recommended (per Kobara)
XmNmarginWidth	8 recommended (per Kobara)
Resources inherited from XmPrimitive:	
XmNbottomShadowColor	Default (Dynamic)
XmNforeground	Black
XmNhighlightThickness	Default (2)
XmNshadowThickness	Default (2)

Table A-2. Resources and settings for Motif widgets (continued).

<u>Resource</u>	<u>Setting</u>
XmNtopShadowColor	Default (Dynamic)
XmNtraversalOn	False
Resources inherited from Core:	
XmNbackground	LightBlue
XmNborderWidth	Default (0)
<u>PushButton (for Default PushButton)</u>	
New resources:	
XmNarmColor	Gray
XmNdefaultButtonShadowThickness	Default (0)
XmNfillOnArm	Default (True)
XmNshowAsDefault	1 recommended (per Kobara)
Resources inherited from XmLabel:	
XmNalignment	Default (XmALIGNMENT_CENTER)
XmNmarginHeight	Default (2) recommended (per Kobara)
XmNmarginWidth	8 recommended (per Kobara)
Resources inherited from XmPrimitive:	
XmNbottomShadowColor	Default (Dynamic)
XmNforeground	Black
XmNhighlightThickness	Default (2)
XmNshadowThickness	Default (2)
XmNtopShadowColor	Default (Dynamic)
XmNtraversalOn	False
Resources inherited from Core:	
XmNbackground	LightBlue
XmNborderWidth	Default (0)
<u>PushButton (in a Menupane)</u>	
New resources:	
XmNfillOnArm	False
Resources inherited from XmLabel:	
XmNalignment	Default (XmALIGNMENT_CENTER)
XmNmarginHeight	Default (2) recommended (per Kobara)
XmNmarginWidth	8 recommended (per Kobara)
Resources inherited from XmPrimitive:	
XmNbottomShadowColor	Default (Dynamic)
XmNforeground	Black

Table A-2. Resources and settings for Motif widgets (continued).

<u>Resource</u>	<u>Setting</u>
XmNhighlightThickness	Default (2)
XmNshadowThickness	Default (2)
XmNtopShadowColor	Default (Dynamic)
XmNtraversalOn	False
Resources inherited from Core:	
XmNbackground	LightBlue
XmNborderWidth	Default (0)
<u>RowColumn (for Menubar)</u>	
New resources:	
XmNadjustLast	False recommended (per Kobara)
XmNadjustMargin	Default (True)
XmNbuttonMnemonics	Default (NULL)
XmNentryBorder	2 recommended (per Kobara)
XmNmenuPost	Default (BSelect Press)
XmNspacing	Default (0) recommended (per Kobara)
Resources inherited from XmManager:	
XmNbottomShadowColor	Default (Dynamic)
XmNshadowThickness	2 recommended (per Kobara)
XmNtopShadowColor	Default (Dynamic)
XmNtraversalOn	False
Resources inherited from Core:	
XmNbackground	LightBlue
XmNborderWidth	Default (0)
<u>RowColumn (for Pulldown Menupane)</u>	
New resources:	
XmNadjustLast	False recommended (per Kobara)
XmNadjustMargin	Default (True)
XmNbuttonMnemonics	Default (NULL)
XmNentryBorder	Default (0)
XmNmargInHeight	2 recommended (per Kobara)
XmNmargInWidth	2 recommended (per Kobara)
XmNmenuPost	Default (NULL)
XmNspacing	0 recommended (per Kobara)
Resources inherited from XmManager:	
XmNbottomShadowColor	Default (Dynamic)
XmNshadowThickness	2 recommended (per Kobara)

Table A-2. Resources and settings for Motif widgets (continued).

<u>Resource</u>	<u>Setting</u>
XmNtopShadowColor	Default (Dynamic)
XmNtraversalOn	False
Resources inherited from Core:	
XmNbackground	LightBlue
XmNborderWidth	Default (0)
<u>RowColumn (for Option Menu)</u>	
New resources:	
XmNadjustLast	False recommended (per Kobara)
XmNadjustMargin	Default (True)
XmNentryBorder	Default (0)
XmNmarginHeight	Default (Dynamic)
XmNmarginWidth	Default (Dynamic)
XmNmenuPost	Default (BSelect Press)
XmNmnemonic	Default (NULL)
XmNspacing	0 recommended (per Kobara)
Resources inherited from XmManager:	
XmNbottomShadowColor	Default (Dynamic)
XmNshadowThickness	2 recommended (per Kobara)
XmNtopShadowColor	Default (Dynamic)
XmNtraversalOn	False
Resources inherited from Core:	
XmNbackground	LightBlue
XmNborderWidth	Default (0)
<u>RowColumn (for Work Area)</u>	
New resources:	
XmNadjustLast	False recommended (per Kobara)
XmNadjustMargin	Default (True)
XmNmarginHeight	Default (3)
XmNmarginWidth	Default (3)
XmNmenuPost	Default (BSelect Press)
XmNspacing	Default (3)
Resources inherited from XmManager:	
XmNshadowThickness	Default (0)
XmNtraversalOn	False

Table A-2. Resources and settings for Motif widgets (continued).

<u>Resource</u>	<u>Setting</u>
Resources inherited from Core:	
XmNbackground	LightBlue
XmNborderWidth	Default (0)
<u>Scale</u>	
New resources:	
XmNhighlightThickness	Default (2)
XmNshowValue	True recommended
XmNtroughColor	SlateGray
Resources inherited from XmManager:	
XmNbottomShadowColor	Default (Dynamic)
XmNforeground	Black
XmNshadowThickness	Default (0)
XmNtopShadowColor	Default (Dynamic)
XmNtraversalOn	False
Resources inherited from Core:	
XmNbackground	LightBlue
XmNborderWidth	Default (0)
<u>ScrollBar</u>	
New resources:	
XmNinitialDelay	Default (250)
XmNrepeatDelay	Default (50)
XmNscrollBarHeight	16 recommended for horizontal Scrollbars (per Kobara)
XmNscrollBarWidth	16 recommended for vertical Scrollbars (per Kobara)
XmNshowArrows	Default (True)
XmNtroughColor	SlateGray
Resources inherited from XmPrimitive:	
XmNbottomShadowColor	Default (Dynamic)
XmNhighlightThickness	2 recommended (per Kobara)
XmNshadowThickness	Default (2) recommended (per Kobara)
XmNtopShadowColor	Default (Dynamic)
XmNtraversalOn	False
Resources inherited from Core:	
XmNbackground	LightBlue
XmNborderWidth	Default (0)

Table A-2. Resources and settings for Motif widgets (continued).

<u>Resource</u>	<u>Setting</u>
<u>ScrolledWindow (for a Scrolled List)</u>	
New resources (for a Scrolled Window):	
XmNscrollBarDisplayPolicy	XmSTATIC recommended (per Kobara)
XmNscrollBarPlacement	XmBOTTOM_RIGHT recommended (per Kobara)
XmNspacing	Default (4) recommended (per Kobara)
New resources (specific to a Scrolled List):	
XmNscrolledWindowMarginHeight	Default (0) recommended (per Kobara)
XmNscrolledWindowMarginWidth	Default (0) recommended (per Kobara)
Resources inherited from XmManager:	
XmNbottomShadowColor	Default (Dynamic)
XmNforeground	Black
XmNshadowThickness	2 recommended
XmNtopShadowColor	Default (Dynamic)
XmNtraversalOn	False
Resources inherited from Core:	
XmNbackground	LightGray
XmNborderWidth	Default (0)
<u>ScrolledWindow (for Editable Text)</u>	
New resources (for a Scrolled Window):	
XmNscrollBarDisplayPolicy	XmSTATIC recommended (per Kobara)
XmNscrollBarPlacement	XmBOTTOM_RIGHT recommended (per Kobara)
XmNspacing	Default (4) recommended (per Kobara)
New resources (specific to Scrolled Text):	
XmNscrolledWindowMarginHeight	2 recommended (per Kobara)
XmNscrolledWindowMarginWidth	2 recommended (per Kobara)
Text-related resources (to be set in the text widget contained in this window):	
XmNautoShowCursorPosition	Default (True)
XmN blinkRate	Default (500)
XmNcursorPositionVisible	Default (True)
XmNeditable	Default (True)
XmNselectionArray	Default XmSELECT_POSITION - resets the insert cursor position XmSELECT_WORD - selects a word XmSELECT_LINE - selects a line of text XmSELECT_ALL - selects all of the text
XmNselectionArrayCount	Default (4)

Table A-2. Resources and settings for Motif widgets (continued).

<u>Resource</u>	<u>Setting</u>
XmNselect Threshold	Default (5)
XmNwordWrap	True recommended (per Kobara)
Resources inherited from XmManager:	
XmNbottomShadowColor	Default (Dynamic)
XmNforeground	Black
XmNshadowThickness	2 recommended
XmNtopShadowColor	Default (Dynamic)
XmNtraversalOn	False
Resources inherited from Core:	
XmNbackground	LightGray
XmNborderWidth	Default (0)
<u>Scrolled Window (for Noneditable Text)</u>	
New resources (for a Scrolled Window):	
XmNscrollBarDisplayPolicy	XmSTATIC recommended (per Kobara)
XmNscrollBarPlacement	XmBOTTOM_RIGHT recommended (per Kobara)
XmNspacing	Default (4) recommended (per Kobara)
New resources (specific to Scrolled Text):	
XmNscrolledWindowMarginHeight	2 recommended (per Kobara)
XmNscrolledWindowMarginWidth	2 recommended (per Kobara)
Text-related resources (to be set in the text widget contained in this window):	
XmNblinkRate	0
XmNcursorPositionVisible	False
XmNeditable	False
Resources inherited from XmManager:	
XmNbottomShadowColor	Default (Dynamic)
XmNforeground	Black
XmNshadowThickness	2 recommended
XmNtopShadowColor	Default (Dynamic)
XmNtraversalOn	False
Resources inherited from Core:	
XmNbackground	Azure2
XmNborderWidth	Default (0)
XmNsensitive	False

Table A-2. Resources and settings for Motif widgets (continued).

<u>Resource</u>	<u>Setting</u>
<u>Separator</u>	
New resources:	
XmNmargin	Default (0) XmNrecommended (per Kobara)
XmNseparatorType	XmSHADOW_ETCHED_IN recommended (per Kobara)
Resources inherited from XmPrimitive:	
XmNbottomShadowColor	Default (Dynamic)
XmNshadowThickness	Default (2) recommended (per Kobara)
XmNtopShadowColor	Default (Dynamic)
XmNtraversalOn	Default (False)
Resources inherited from Core:	
XmNbackground	LightBlue
XmNborderWidth	Default (0)
<u>Text (Editable)</u>	
New resources:	
XmNautoShowCursorPosition	Default (True)
XmNblinkRate	Default (500)
XmNcursorPositionVisible	Default (True)
XmNeditable	Default (True)
XmNselectionArray	Default XmSELECT_POSITION - resets the insert cursor position XmSELECT_WORD - selects a word XmSELECT_LINE - selects a line of text XmSELECT_ALL - selects all of the text
XmNselectionArrayCount	Default (4)
XmNselectThreshold	Default (5)
XmNwordWrap	True recommended (per Kobara)
Resources inherited from XmPrimitive:	
XmNbottomShadowColor	Default (Dynamic)
XmNforeground	Black
XmNhighlightThickness	Default (2)
XmNshadowThickness	Default (2)
XmNtopShadowColor	Default (Dynamic)
XmNtraversalOn	False
Resources inherited from Core:	
XmNbackground	LightGray
XmNborderWidth	Default (0)

Table A-2. Resources and settings for Motif widgets (continued).

<u>Resource</u>	<u>Setting</u>
<u>Text (Noneditable)</u>	
New resources:	
XmN blinkRate	0
XmNcursorPositionVisible	False
XmNeditable	False
Resources inherited from XmPrimitive:	
XmNbottomShadowColor	Default (Dynamic)
XmNforeground	Black
XmNhighlightThickness	Default (2)
XmNshadowThickness	Default (2)
XmNtopShadowColor	Default (Dynamic)
XmNtraversalOn	False
Resources inherited from Core:	
XmNbackground	Azure2
XmNborderWidth	Default (0)
XmNsensitive	False
<u>TextField (Containing Editable Text)</u>	
New resources:	
XmNblinkRate	Default (500)
XmNcursorPosition	Default (0)
XmNcursorPositionVisible	Default (True)
XmNeditable	Default (True)
XmNmarginHeight	2 recommended (per Kobara)
XmNmarginWidth	Default (3) recommended (per Kobara)
XmNselectionArray	Default XmSELECT_POSITION - resets the insert cursor position XmSELECT_WORD - selects a word XmSELECT_LINE - selects a line of text
XmNselectionArrayCount	Default (3)
XmNselectThreshold	Default (5)
Resources inherited from XmPrimitive:	
XmNbottomShadowColor	Default (Dynamic)
XmNforeground	Black
XmNhighlightThickness	Default (2)
XmNshadowThickness	Default (2)
XmNtopShadowColor	Default (Dynamic)
XmNtraversalOn	False

Table A-2. Resources and settings for Motif widgets (continued).

<u>Resource</u>	<u>Setting</u>
Resources inherited from Core:	
XmNbackground	LightGray
XmNborderWidth	Default (0)
<u>TextField (Containing Noneditable Text)</u>	
New resources:	
XmNblinkRate	0
XmNcursorPositionVisible	False
XmNeditable	False
XmNmarginHeight	2 recommended (per Kobara)
XmNmarginWidth	Default (5)
Resources inherited from XmPrimitive:	
XmNbottomShadowColor	Default (Dynamic)
XmNforeground	Black
XmNhighlightThickness	Default (2)
XmNshadowThickness	Default (2)
XmNtopShadowColor	Default (Dynamic)
XmNtraversalOn	False
Resources inherited from Core:	
XmNbackground	Azure2
XmNborderWidth	Default (0)
XmNsensitive	False
<u>ToggleButton (in a RadioBox)</u>	
New resources:	
XmNfillOnSelect	Default (True)
XmNindicatorOn	Default (True)
XmNindicatorSize	Default (XmINVALID_DIMENSION)
XmNindicatorType	XmONE_OF_MANY if a Radio Button; XmN_OF_MANY if a Check Button
XmNselectColor	SlateGray
XmNspacing	Default (4) recommended (per Kobara)
XmNvisibleWhenOff	Default (True)
Resources inherited from XmLabel:	
XmNalignment	Default (XmALIGNMENT_CENTER)
XmNlabelType	Default (XmSTRING) recommended (per Kobara)
XmNmarginHeight	Default (2) recommended (per Kobara)
XmNmarginWidth	Default (2) recommended (per Kobara)

Table A-2. Resources and settings for Motif widgets (continued).

<u>Resource</u>	<u>Setting</u>
Resources inherited from XmPrimitive:	
XmNbottomShadowColor	Default (Dynamic)
XmNforeground	Black
XmNhighlightThickness	Default (2)
XmNshadowThickness	Default (Dynamic)
XmNtopShadowColor	Default (Dynamic)
XmNtraversalOn	False
Resources inherited from Core:	
XmNbackground	LightBlue
XmNborderWidth	Default (0)
<u>ToggleButton (in a Menupane)</u>	
New resources:	
XmNfillOnSelect	Default (True)
XmNindicatorOn	Default (True)
XmNindicatorSize	Default (XmINVALID_DIMENSION)
XmNindicatorType	XmONE_OF_MANY if a Radio Button; XmN_OF_MANY if a Check Button
XmNselectColor	SlateGray
XmNspacing	Default (4) recommended (per Kobara)
XmNvisibleWhenOff	Default (False)
Resources inherited from XmLabel:	
XmNalignment	Default (XmALIGNMENT_CENTER)
XmNlabelType	Default (XmSTRING) recommended (per Kobara)
XmNmarginHeight	Default (2) recommended (per Kobara)
XmNmarginWidth	Default (2) recommended (per Kobara)
XmNmnemonic	Default (NULL)
Resources inherited from XmPrimitive:	
XmNbottomShadowColor	Default (Dynamic)
XmNforeground	Black
XmNhighlightThickness	Default (2)
XmNshadowThickness	Default (Dynamic)
XmNtopShadowColor	Default (Dynamic)
XmNtraversalOn	False
Resources inherited from Core:	
XmNbackground	LightBlue
XmNborderWidth	Default (0)

APPENDIX B USER INTERFACE SPECIFICATIONS CHECKLIST

The checklist provided in this appendix was developed as a reference aid for developers as they implement the specifications in their applications. As indicated in section 1, developers should use the checklist as a supplement to, rather than substitute for, the detailed provisions presented in the document.

The checklist contains one-sentence, single-line statements of the provisions included in sections 2 through 8 of the specifications. The items in the checklist are grouped according to the section in which the provisions appear and are numbered sequentially based on the order in which they occur in the document. Items that are mandatory provisions are labeled "M;" items that are recommended provisions are labeled "R." In addition, the checklist includes some items that describe optional provisions (e.g., contain "may") or are worded in the negative (e.g., "keyboard interchangeability is not required"); these items do not include a label.

The checklist, which was developed by Richard Obermayer, Pacific Science and Engineering Group, San Diego, CA, is available in electronic form from Kathleen Fernandes, NRaD, Code 423, San Diego, CA 92152.

USER INTERFACE SPECIFICATIONS CHECKLIST

SECTION 2. USER-COMPUTER INTERACTION

Section 2.1 Input Devices

- ___ M 1 Users shall make selections and control computer interaction by direct manipulation.
- ___ M 2 A pointing device shall be the primary means of interaction, allowing users to navigate and select.
- ___ M 3 The keyboard shall be used for text entry and shall be available for use with the pointing device.
- ___ R 4 Users should interact with the input device that is most appropriate to the task being performed.
- ___ 5 Applications can support multiple input devices for performing the same action.
- ___ R 6 If a window requires different input devices, the movement between devices should be minimized.
- ___ R 7 If multiple input devices are provided for an action, do it consistently for all cases of the action.
- ___ R 8 The keyboard should permit critical operations (e.g., saving data) if the pointing device fails.

Section 2.1.1.1 The Pointer

- ___ M 1 The pointing device shall be associated with a single pointer on the screen.
- ___ M 2 The hotspot of the pointer shall indicate the precise location where operations occur.
- ___ M 3 Users shall move the pointer by moving the pointing device; the pointer shall move anywhere.
- ___ M 4 When users move the pointing device, the pointer shall move in the corresponding direction.
- ___ R 5 For most user interactions, the pointing device-to-pointer movement ratio should be close to 1:1.
- ___ R 6 The pointer should remain in place until moved; the pointer should not be moved by an application.
- ___ R 7 The pointer should not deviate more than .05 inch in any direction; .01 inch for high stability.
- ___ M 8 For multiple screens, the pointer shall move between screens by moving the pointing device.
- ___ M 9 The pointer shall not move beyond the screen boundaries or disappear from sight.
- ___ M 10 The location of the hotspot shall not move as the pointer changes from one shape to another.

Section 2.1.1.2 Pointer Shapes

- ___ M 1 The shape of the pointer shall depend on the functionality of the area in which it is located.
- ___ M 2 The pointer shapes listed in table 2-1 shall be used for the functionality indicated.
- ___ M 3 Applications shall redefine the pointer shape only within an application window.
- ___ M 4 The upper-left-pointing arrow shall be used for object selection in most windows.
- ___ M 5 The X pointer shape shall not be used by an application.
- ___ R 6 New shapes should not be created to symbolize the functionality in table 2-1.
- ___ R 7 Shapes in table 2-1 should not be used to symbolize a different functionality.
- ___ R 8 New pointer shapes should be easy to see, with a hotspot that is obvious and easy to locate.
- ___ R 9 New shapes should suggest their purpose and should not be confused with other objects.

Section 2.1.1.3 Pointing Device Buttons

- ___ M 1 Each button on the pointing device shall have a specific function that is always executed.
- ___ M 2 The Left (L) button shall be used to select an object; the Right (R) button to display a pop-up menu.
- ___ R 3 Left-handed users should be able to exchange the functions between the R button and the L button.
- ___ M 4 A function assigned to the Middle button must not contradict/interfere with another assignment.

Section 2.1.2.1 Keyboard Functions

- ___ R 1 Keyboard should enter text, move the text cursor, select from menu, do default action, and search.
- ___ R 2 Keystrokes made during text entry should appear at the location of the text cursor.
- ___ R 3 Keystrokes made for non-text entry should not be displayed.

Section 2.1.2.2 The Text Insertion Cursor

- ___ M 1 When a window has input focus, the text cursor shall indicate where typed text will be displayed.
- ___ M 2 The text cursor shall appear in the first text entry area, before any text appearing in the area.

Version 1.1, Change 1

- ___ M 3 The standard text cursor shape shall be a flashing pipe (|).
- ___ M 4 The flash rate for the text cursor shall be 2-5 Hz.
- ___ M 5 When a window loses input focus, the text cursor shall remain visible but stop flashing.
- ___ M 6 When a window regains focus, the text cursor shall appear at its previous location and flash.
- ___ R 7 When the pointer is in a text entry area, it should appear as an I-beam shape.
- ___ R 8 When the L button is clicked for text entry, the text cursor should appear and the pointer stays.
- ___ R 9 When users begin typing, the pointer should disappear from the screen.
- ___ R 10 The pointer should reappear when users stop typing or when they move the pointing device.
- ___ R 11 The pointer should change to an I-beam shape only in an area where text entry is possible.
- ___ R 12 The text cursor should appear only in text entry areas, and not where text entry is not possible.
- ___ R 13 Text entry should be possible only after the text cursor is visible at a legal location.
- ___ R 14 Text entry should not be accepted by an application when the text cursor is not visible.
- ___ R 15 Applications should ensure that the text cursor is highly visible.

Section 2.1.2.3 Navigation Key Assignments

- ___ M 1 When a navigation function moves the text cursor, it shall appear at the left of the text field.
- ___ M 2 The Tab key shall move the text cursor to the next single-line text entry field.
- ___ M 3 The Tab key shall move the text cursor to the next tab stop in a multiple-line text field.
- ___ M 4 The direction of cursor movement shall be from left to right, top to bottom.
- ___ M 5 At the bottom-rightmost field, the Tab key shall move the text cursor to the top-leftmost field.
- ___ M 6 The Shift and Tab keys shall be used together to perform backward tabulation.
- ___ M 7 The Down Arrow key shall move the text cursor to the field below the current one.
- ___ M 8 The Up Arrow key shall move the text cursor to the field above the current one.
- ___ M 9 The Left Arrow shall move to the choice to the left; the Right Arrow to the choice to the right.
- ___ M 10 If the text cursor reaches top-left, top-right, bottom-left or bottom-right, the cursor shall wrap.
- ___ M 11 Scrolling shall be available so the cursor remains visible when an arrow key is pressed.
- ___ M 12 If the text cursor is at the bottom of a scrollable area, pressing an arrow key shall make it visible.
- ___ M 13 When Delete key is pressed, the character to the right shall be deleted.
- ___ M 14 When Backspace key is pressed, the character to the left shall be deleted.
- ___ M 15 If more than one character is selected, Delete or Backspace shall delete them all.
- ___ M 16 The Spacebar shall insert a space character.
- ___ M 17 In a multi-line field, Enter shall move the text cursor to the beginning of the next line.
- ___ M 18 When not in a multi-line field, Enter shall invoke the default action in the window.

Section 2.1.2.4 Modifier Key Assignments

- ___ M 2 Shift shall be used with the pointing device for object selection/deselection.
- ___ M 3 Control and Alt shall be used with the pointing device to drag objects to a new location.
- ___ M 4 Shift, Alt and Control, with other keys, shall define keyboard accelerators to select menu options.
- ___ M 5 Standard keyboard accelerators shall be selected from table 4-1.

Section 2.1.2.5 Function Key Assignments

- ___ M 1 The F1 function key shall provide access to context-specific help.

Section 2.2.1 Object-Action Selection

- ___ R 1 Applications should base their interactions with users on an object-action selection paradigm.

Section 2.2.2 Selecting Objects

- ___ M 1 The methods in table 2-2 shall be used to select or deselect text and graphics objects.
- ___ M 2a To select one or more characters, click on the first, drag to the last, then release.
- ___ M 2b The characters shall highlight as the pointer is dragged over them.
- ___ M 3 If the pointer is dragged over several lines, text to right and left of the pointer shall highlight.

Version 1.1, Change 1

- ___ M 4a To add contiguous text, press Shift and click on last character to be added.
- ___ M 4b The text in existing selection plus added text shall highlight.
- ___ M 5a To deselect text, click on another object or click in an empty screen area.
- ___ M 5b When text is deselected, it shall return to its normal appearance.
- ___ M 6a To deselect one or more contiguous characters, press Shift, then click on text.
- ___ M 6b When text is deselected, text to end shall return to normal appearance; rest remains highlighted.
- ___ M 7 To select single graphic object, click on the object; it shall highlight.
- ___ M 8a To select a group of contiguous graphic objects, drag from first to last, then release.
- ___ M 8b A box shall outline objects selected; release L button, box disappears, and objects highlight.
- ___ M 9a To add a graphic object to a selection, press Shift, and click on the object; it also highlights.
- ___ M 9b The previously selected objects remain highlighted throughout the selection process.
- ___ M 10a To add a group of contiguous graphic objects, press Shift, drag first to last, then release.
- ___ M 10b As pointer is dragged, a bounding box is displayed outlining objects being selected.
- ___ M 10c When Shift and L button are released, box disappears, and outlined objects highlight.
- ___ M 10d The objects in the existing selection remain highlighted throughout the selection process.
- ___ M 11 To deselect a graphic object, click outside the object or on another object; it returns to normal.
- ___ M 12 When a new object is selected, the previously selected object is deselected.
- ___ M 13 To deselect an object in a group, press Shift, click on the object; its appearance returns to normal.
- ___ M 14a To deselect contiguous objects in a group, press Shift, and drag over the objects being deselected.
- ___ M 14b A bounding box outlines the objects; when Shift and L button are released, the box disappears.
- ___ M 14c The objects that were in the box return to normal; the others in the group remain highlighted.
- ___ R 15 Users, where appropriate, should be able to quickly select or deselect all objects in a window.
- ___ R 16 Methods should be available to select objects quickly and efficiently within the application.
- ___ R 17 For an overlapping group, a method should be implemented to quickly select the desired object.

Section 2.2.3 Other Types of Selection

- ___ M 1 When a default action is assigned, the action shall be executed by double clicking on the object.
- ___ M 2 A default in a window shall be executed by double clicking or by pressing Return after selecting.
- ___ R 3 Extended selection should be available for text objects.
- ___ R 5 Extended selection: Click places text cursor, double click selects word, triple click selects line.

Section 2.2.4 Manipulating Objects

- ___ R 1 Objects should be manipulated by dragging them or by editing them using a clipboard.
- ___ R 3 Access to commands should be provided in a consistent fashion.
- ___ R 4 Users should be able to view the contents of the clipboard buffer.
- ___ R 5 Users should be informed when cutting or copying an object whose size exceeds the buffer.

Section 2.2.4.1 Dragging Objects to a New Location

- ___ R 1 Users should be able move or copy the object(s) they have selected by dragging them.
- ___ R 2 Copy: Press Control, drag object to new location, release to paste; original highlights and remains.
- ___ R 3 Move: Press Alt, drag object to new location, release to paste; original is deleted.
- ___ R 4 These methods of manipulation should be used to transfer objects from one window to another.
- ___ R 5 Dragging from a noneditable component should be treated as a copy.
- ___ R 6 Dragging to a transport device (printer) should be treated as a copy.
- ___ R 7 Dragging to a removal device (wastebasket) should be treated a move.
- ___ R 8 Dragging text to/from a text collection should be treated as a move.
- ___ R 9 Otherwise, dragging an object should be treated as a move.

Section 2.2.4.2 Using Editing Commands

- ___ R 1 Users should have the following editing commands: Cut, Copy, Paste, Clear, Delete, Undo.
- ___ R 2a Cut clears the clipboard, stores a copy of the object in the clipboard, removes object from window.

Version 1.1, Change 1

- ___ R 2b If cut object is graphic, previous space is left blank; if text, remaining text is compressed.
- ___ R 3 Copy clears the clipboard and stores a copy of object in clipboard; object remains.
- ___ R 4a Paste copies the object in the clipboard to a new location.
- ___ R 4b If object is text, paste copies the object to text cursor location; text appears to left of cursor.
- ___ R 4c If object is graphic, paste copies to pointer location in window with input focus.
- ___ R 4d Pasted object remains in the clipboard to be used for multiple pastes.
- ___ R 5 Clear removes object from window without copying to clipboard; previous space is not compressed.
- ___ R 6 Delete removes object from window without copying to clipboard; previous space is compressed.
- ___ R 7 Undo reverses the most recently executed action.
- ___ R 8 Editing commands should be provided as push buttons or pull-down menus.
- ___ R 9 Editing commands should do transfer within window, to other windows, and between applications.

Section 2.3 Text Entry

- ___ R 1 Applications should be designed so that insert mode is used for text entry.
- ___ R 2 Clicking once should position the text cursor at the location of the pointer.
- ___ R 3 New text should be added at the insertion point, and cursor plus existing text should move right.
- ___ R 4 To delete the character to the left of the text cursor, users should press the Backspace key.
- ___ R 5 To delete the character to the right of the text cursor, users should press the Delete key.
- ___ R 6 Double clicking on text should select the word at the location of the pointer.
- ___ R 7 Dragging the pointer across text should also select it.
- ___ R 8 Upon typing, the highlighted text should disappear, and the text cursor plus new text is inserted.
- ___ R 9 For highlighted text, Delete should make text disappear, cursor appear, and other text compress.

Section 2.4.1 User Control of Interaction

- ___ R 1 Applications should execute an action only in response to explicit user input.
- ___ R 2 Users should be able to specify when a process occurs.
- ___ R 3 Users should be able to take actions that will interrupt or terminate a process.

Section 2.4.2 Immediate Feedback

- ___ R 1 Users should receive immediate feedback on the result of their action.
- ___ R 2 When users take an action, there should be an immediate and visible response to the action.
- ___ R 3 A visible response to the action should occur even if the result cannot be displayed immediately.
- ___ R 4 Applications should inform users when they can and cannot take an action.
- ___ R 5 Visual cues should show if an application is: accepting input, temporarily unavailable, unavailable.
- ___ R 6 The appearance of an object should give an indication of its availability.
- ___ R 7 If users must do several actions for an operation, the users should be prompted for the actions.
- ___ R 8 Applications should ignore user actions made during periods when input cannot be accepted.
- ___ R 9 Keystrokes should be accepted only when the text cursor appears in a window with input focus.
- ___ R 10 Applications should disable the pointing device and/or keyboard if input may be destructive.
- ___ R 11 Users should not be able to override disabling but should be able to stop a process.

Section 2.4.3 Response Time

- ___ R 1 Visible response should be within .2 seconds; data display should take no longer than .5 - 1.0 sec.
- ___ R 2 Requests for new displays may take 2 - 10 sec.
- ___ R 3 A visible response should be made within .1 sec.; notify users if more than 5 sec. is needed.
- ___ R 4 Error feedback should be provided to users within 2 sec. of detection.
- ___ R 5 If a request takes more than 2 sec., the pointer shape should be changed to an hourglass.
- ___ R 6 If a request requires more than 5 sec., a message should indicate that the operation is lengthy.

Section 2.4.4 Error Detection

- ___ R 1 If users attempt an invalid action, the application should only display a message.

Version 1.1, Change 1

- ___ R 2 If users make multiple errors with a single action, they should be notified of each error.
- ___ R 3 Feedback should be immediate, be visual and/or auditory, and explain the nature of the error.
- ___ R 4 If an error is repeated, the feedback should show that the attempted correction was processed.
- ___ R 5 Users should be required to correct only the invalid action and not to repeat the entire sequence.
- ___ R 6 After correcting, users should execute the same action for re-entry that was used originally.

Section 2.4.5 Explicit Destruction

- ___ M 1 Users shall confirm destructive actions before the action is executed.
- ___ M 2 Users shall respond to a confirmation prompt for any action that cannot be reversed or "undone."
- ___ R 3 If destructive action applies to multiple objects, a list should be shown to select the ones that apply.
- ___ R 4 Users should not have to confirm when they close a window unless the action will cause data loss.

Section 2.4.6 General "Undo" Capability

- ___ R 1 Users should be able to "undo" the most recent selection/action except for explicit destruction.
- ___ R 2 Undo should be able to deselect, return to a prior state, and retrieve previous screen information.
- ___ R 3 Irreversible actions should be labeled and should be clearly separated from those that are not.

Section 2.5.1 Consistency at the Application Level

- ___ R 1 Sequences of actions should allow completion rapidly and with a minimum number of actions.
- ___ R 2 Each operation should be achievable with a consistent standard action sequence.
- ___ R 3 Other approaches can be provided, but the standard way should always be available.
- ___ R 4 Users should have to execute only those actions that are required to perform an operation.

Section 2.5.2 Consistency at the System Level

- ___ R 1 Users should perform an operation the same way in different applications within a system.

SECTION 3. WINDOWS

Section 3.1 Window Components

- ___ M 1 A primary window shall contain all of the standard window components.
- ___ M 2 A secondary window shall contain a title area and a Window Menu button.
- ___ M 3 A secondary window may contain resize borders and Maximize button if window should be resized.
- ___ M 4 The Window Menu in a secondary window shall contain options for the window components included.
- ___ M 5 A secondary window shall not include a Minimize option in its Window Menu.
- ___ M 6 Lower in a secondary window shall lower all of the children of the window's parent.
- ___ M 7 A secondary window shall not have a Minimize button.
- ___ M 8 Each window shall contain only components that are appropriate and shall behave consistently.
- ___ M 9 Each window should contain only one set of window components that are selectable by users.
- ___ M 10 If non-Motif windows are used, only one set of window components should be available.

Section 3.2.1 Input Focus

- ___ M 1 Only one window shall have input focus (i.e., receive input from the keyboard) at any time.
- ___ M 2 A window shall have input focus when it first appears on the screen.
- ___ M 3 Users shall assign input focus by moving the pointer into the window and clicking the L button.
- ___ M 4 Clicking in an empty part of the window shall raise it to the front, and the frame shall highlight.
- ___ M 5 If users click on an object, the window shall raise to front, frame highlight, and object be selected.
- ___ M 6 If the window with input focus is closed, users shall explicitly select another window to have focus.
- ___ M 7 An application shall not arbitrarily assign focus to another window on the screen.

Section 3.2.2 Open, Active, and Closed Windows

- ___ M 1 Multiple windows may be open at any one time and may partially or totally obscure one another.

Version 1.1, Change 1

- ___ M 2 An active window's content shall be updated continuously, not when just it regains input focus.
- ___ M 3 When a window is closed, it shall be removed from the screen.
- ___ M 4 In most cases, all processing relating to the closed window shall stop.

Section 3.2.3 Iconified Windows

- ___ M 1 When a window is iconified, it shall be replaced with an icon at the screen's lower left margin.
- ___ M 2 Any processing that was occurring in the window shall continue while the window is iconified.
- ___ M 3 When the icon is opened, the window shall be displayed in front, where it was before.

Section 3.3.1 Placing a Window

- ___ M 1 When a window is closed and then reopened, it shall be displayed where it was when closed.

Section 3.3.2 Moving a Window

- ___ M 1 To move a window, users shall select Move option in the Window Menu or drag the title bar.
- ___ M 2a When the Move option is selected, a four-directional arrow appears.
- ___ M 2b An outline of the window shall move on the screen as users move the pointer.
- ___ M 3a With the drag method, users drag the title bar, and the window moves to a new location.
- ___ M 3b An outline of the window moves on the screen as the pointer is moved.
- ___ M 3c Releasing the L button places the window in its new location.

Section 3.3.3 Resizing a Window

- ___ M 1 To expand a window to maximum, users shall select Maximize button or Window Menu option.
- ___ M 2 To return a maximized window to previous size, users shall select Maximize button or Restore option.
- ___ M 3 To change the size of a window, users shall select the Size option or drag the resize border.
- ___ M 4a When the Size option is selected, the four-directional arrow appears.
- ___ M 4b When the pointer reaches the border, it changes to a resize pointer.
- ___ M 4c As pointer is moved, an outline of the window moves; releasing L button redraws window.
- ___ M 5a With the drag method, pointer is positioned on resize border, and pointer changes to resize pointer.
- ___ M 5b The L Button is pressed, and the window dragged to the desired size.
- ___ M 5c The window is resized when users release the L button (the pointer returns to its normal shape).

Section 3.3.4 Iconifying a Window

- ___ M 1 A window shall be iconified by selecting Minimize button or Window Menu option.
- ___ M 2 The icon appears in the lower left corner of the screen and shall contain a graphic image and label.
- ___ M 3 If more than one window is iconified, the icons are arrayed in a row from left to right.
- ___ M 4 The label shall be located below the image and have the same title as its corresponding window.
- ___ M 5 When a window is iconified, any ongoing processing associated with the window continues.
- ___ M 6 Icons can be moved using the same methods as for moving windows.
- ___ M 7 To open an icon, users point to the icon and double click the L button.

Section 3.3.5 Lowering a Window

- ___ M 1 To lower a window to the bottom of the hierarchy, select the Lower option in the Window Menu.

Section 3.3.6 Closing a Window

- ___ M 1 To close, users shall select Close from Window Menu or double click on Window Menu button.
- ___ M 2 If processing is occurring or data are unsaved, users shall be required to confirm before closing.

Section 3.4.1 Primary Windows

- ___ M 1 A hierarchy of primary and secondary windows shall be defined to provide the functionality.
- ___ M 2 If a primary window is iconified, it and all its secondary windows shall be replaced with an icon.
- ___ M 3 When a primary window is iconified, all processing in the window shall continue.

Version 1.1, Change 1

- ___ M 4 When an icon is opened, the window and its secondary windows that were displayed shall re-appear.
- ___ M 5 Each primary window in an application shall be iconifiable separately.
- ___ M 6 When a primary window is closed, it and its secondary windows shall be removed from the screen.
- ___ M 7 When a primary window is closed, all processing in it shall stop (except as noted in section 3.2.2).
- ___ M 8 When the last primary window for an application is closed, the application shall also be closed.

Section 3.4.2.1 Dialog Windows

- ___ M 1 When a secondary window (e.g., dialog window) is opened, it shall appear in front of its parent.
- ___ R 2 When a modeless dialog window appears, users shall be able to interact with other windows.
- ___ R 3 When a modal dialog window appears, users shall not interact with application/system windows.
- ___ R 4 If a secondary window is closed, its children shall be closed but its parent shall not be affected.

Section 3.4.2.2 Message Windows

- ___ R 1 Users should be able to move a message window but not iconify or resize it.

SECTION 4. MENUS

Section 4.1.1 Menu Title

- ___ R 1 The title of a menu should describe the category or type of options and be different from other titles.
- ___ R 2 The first letter of each word in the menu title should be capitalized.
- ___ R 3 If the title contains an acronym, it should be capitalized.

Section 4.1.2.1 Types of Options

- ___ M 1 Routing options that display a window shall be followed by an ellipsis.
- ___ M 2 Routing options that display a cascading menu shall be followed by a right-pointing arrow.
- ___ R 3 Options that are action toggles should be worded to reflect the action.
- ___ R 4 The wording of toggle options should be semantically congruent with natural usage.
- ___ R 5 When users select a toggle, its wording should change to reflect when the action is selected again.
- ___ M 6 Only one of the action toggles shall appear in the menu at any time.
- ___ R 7 For an "undo" option, the wording should change dynamically to reflect the action to be undone.
- ___ R 8 State toggle wording should describe the state and may include radio or check buttons to the left.
- ___ R 9 If users select one of these options, the button should highlight without the wording changing.

Section 4.1.2.2 Wording of Options

- ___ R 1 Menu options should be phrased to reflect the action executed, and worded in user vocabulary.
- ___ R 2 The vocabulary in table 5-1 should be used if the actions listed in the table are included.
- ___ R 3 Options should be tersely worded and in upper/lower case letters, with first letter capitalized.
- ___ R 4 All acronyms should be capitalized.
- ___ R 5 Each option should be left-justified and appear on a single line.
- ___ R 6 Long menu options should be in a wide menu rather than extending the item beyond one line.

Section 4.1.2.3 Organization and Grouping

- ___ R 1 Pull-down menus should have logical or functional groupings of options, with clear titles.
- ___ R 2 If not in logical groups, options should be ordered by frequency of usage, most frequent at the top.
- ___ R 3 Less frequently executed options and destructive options should be at the bottom of the menu.
- ___ R 4 If similar options are in different menus, the options should be ordered in a consistent manner.
- ___ R 5 If logical groups or frequency is inappropriate, options may be in alphabetical or numerical order.
- ___ R 6 Whenever possible, a menu should contain no more than ten options.
- ___ R 7 Menus with less than three options should be avoided.
- ___ R 8 If a menu has over four options, divide into groups (no more than 4 options) separated with a line.

Version 1.1, Change 1

Section 4.1.2.4 Available and Unavailable Options

- ___ R 1 If an option or or set of options is never available to a user, the option(s) should not be in a menu.
- ___ R 2 If an option is temporarily unavailable, it should be displayed but dimmed.
- ___ R 3 The wording of options may change (e.g., when an option is an action toggle).
- ___ R 4 Options should not be added to or deleted from a menu to indicate their availability.

Section 4.1.3 Keyboard Accelerators

- ___ M 1 Applications shall use keyboard accelerators to provide keyboard access to frequently used options.
- ___ R 2 If an accelerator is available, it should be right justified and separated from the option.
- ___ M 3 Users shall type the keyboard accelerator for a menu option to execute the option.
- ___ M 4 Users shall be able to execute keyboard accelerators for the window with input focus.
- ___ R 5 If a keyboard accelerator contains an alphabetic character, it should not be case sensitive.
- ___ R 6 Keyboard accelerators listed in table 4-1 should be implemented to standardize the accelerators.

Section 4.1.4 Cascading Menus

- ___ R 1 Cascading menus should present actions hierarchically and shorten menus over 10-15 options.
- ___ M 2 Cascading menus shall appear when the parent option is selected or pointer is dragged on the option.
- ___ M 3 Cascading menus shall be to the right of the parent menu (below if space to the right is limited).
- ___ R 4 The use of an excessive number of cascading menus should be avoided.
- ___ R 5 Cascading menus should be limited to three levels.
- ___ R 6 New menus or a dialog window should be created if the number of cascading menus becomes too large.

Section 4.1.5 Menu Selection Methods

- ___ M 1 To browse the menus in a menu bar, users shall press the L button and drag across the titles.
- ___ M 2 Each menu shall appear as the pointer crosses its title.
- ___ M 3 When a menu is displayed, the first available option in the menu shall highlight.
- ___ M 4 Two methods shall open a menu and select an option: Drag Method and Click Method.
- ___ M 5 With drag method, options shall highlight and cascade menus display as pointer moves over them.
- ___ M 6 With click method, click L on menu; menu is displayed ("sticky"), and first item shall highlight.
- ___ M 7 To "unstick" (close) the menu, move pointer off menu and click.
- ___ R 8 Tear-off menus should be included which remain displayed to make multiple selections.

Section 4.2 Pop-Up Menus

- ___ 1 Pop-up menus can contain the same components as a pull-down menu including keyboard accelerators.
- ___ M 2 To display a pop-up menu (where they are available), users shall press the R button.
- ___ M 3 Pop-ups shall disappear after selection (by releasing the R button) or if pointer moves off the menu.
- ___ R 4 Pop-up menus should appear on the screen near the object being manipulated (via pointer location).
- ___ R 5 A window containing a pop-up menu should provide some indication that the menu is available.
- ___ R 6 A pop-up menu should not contain options that duplicate other controls that are already available.

Section 4.3 Option Menus

- ___ M 1 The title of an option menu shall be a label that appears next to or above the option button.
- ___ M 3 The option button shall be followed by a bar graphic to distinguish it from an action push button.
- ___ M 4 When users point to the option button and press the L button, the option menu shall appear.
- ___ M 5 Users shall drag the pointer to the desired option and release the L button to select the option.
- ___ M 6 The menu shall disappear, and the option that was selected shall be displayed in the option button.

SECTION 5. CONTROLS

Section 5.1 Push Buttons

- ___ M 1 A push button shall be used to initiate an action.

Version 1.1, Change 1

- ___ M 2a To select a push button, users shall point to the button and click the L button.
- ___ M 2b When the Left button is pressed, the push button shall highlight.
- ___ M 2c When L button is released, appearance becomes normal, and the action shall be executed.
- ___ M 3 To not execute, users shall move the pointer off while L button is pressed and then release.
- ___ M 4 For a default push button, users shall double-click on it or press Return after making selection.

Section 5.1.1 Standard Vocabulary

- ___ R 1 Push button labels should be short and unambiguous.
- ___ R 2 Action buttons labels should describe and reflect the action taken by the application.
- ___ R 3 For OK/Yes response buttons, the prompt should be phrased carefully and unambiguously.
- ___ R 4 "All" should be used in push button labels only when there is no ambiguity as to referent.
- ___ R 5 Push button labels where multiple referents possible should use the name of the object/element.
- ___ R 6 Button names should indicate the referred object, and the buttons should be near it.
- ___ R 7 Application developers should use the vocabulary in table 5-1 whenever possible.
- ___ R 8 Developers can create new vocabulary as needed to describe actions not listed in the table.
- ___ R 9 An action should be a verb, in active voice, and describe what the button does.
- ___ R 10 The names of actions should be congruent (e.g., save/delete, on/off, in/out).
- ___ R 11 New terms should not be created for table 5-1 actions; these terms should not be used differently.
- ___ R 12 Developers should ensure that a label always describes the same action.

Section 5.1.2 Default Push Button Actions

- ___ R 1 Default actions should be identified in dialog windows, and a default push button designated.
- ___ R 2 A default push button has an extra border, and can be activated by pressing Return.
- ___ R 3 The default button should appear in the same position within the set of push buttons.
- ___ R 4 OK is frequently the default push button for dialog windows.
- ___ R 5 OK and Yes should not be labels for a default push button if the action is potentially destructive.
- ___ R 6 Actions performed by the default push button should be reversible.
- ___ R 7 If there is no default push button, pressing Return should have no effect.

Section 5.1.3 Modifying Normal Push Button Behavior

- ___ R 1 Normal push button behavior: Highlight when L-button pressed, normal when released.
- ___ R 2 To serve as indicator: Remain highlighted when selected, normal when another is selected.
- ___ R 3 Clicking the heading of multi-column list does sort; heading button of sorted column highlights.

Section 5.2 Radio Buttons

- ___ M 1 Radio buttons shall be used for selecting one from at least two mutually exclusive options.
- ___ M 2a To select a radio button, users shall point to the button or the label and click the L button.
- ___ M 2b The radio button shall highlight when selected.
- ___ M 2c If a radio button is selected, any selected button in the group shall be deselected.
- ___ M 3 Users shall deselect a radio button by selecting it or a different button in the same group.

Section 5.3 Check Buttons

- ___ M 1 A check button shall be used to set options in an application.
- ___ M 2 A check button is a nonexclusive setting; selecting a check button shall toggle a setting or state.
- ___ M 3 Check buttons can be used singly or in related groups.
- ___ M 4a To select a check button, users shall point to the button or the label and click the L button.
- ___ M 4b The check button shall highlight when selected.
- ___ M 4c When a check button is selected, any other selected check buttons shall remain selected.
- ___ M 5 Users shall be able to select any or all of the check buttons in a group.
- ___ M 6 Users shall deselect a check button by selecting it again.

Version 1.1, Change 1

Section 5.4 Text Entry Fields

- ___ M 1 A text entry field shall include a title and a text entry area.
- ___ M 2 The title shall appear to the left or above the entry and describe what is to be entered.
- ___ 3 An entry area may include scroll bars for text that is long or extends beyond a single line.

Section 5.4.1 Format of Text Entry Fields

- ___ R 1 The title of a text entry field should include cues regarding format.
- ___ R 2 If a unit of measurement is always used, it should be in the title and not have to be entered.
- ___ R 3 The title should be designed so that users can easily distinguish it from the text entry area.
- ___ R 4 The text entry area should indicate the basic features of the entry required.
- ___ R 5 The entry area should be long enough for users to enter the information required.
- ___ R 6 If an entry is a fixed length, then the entry area should be the same length.
- ___ R 7 If entry length varies, the entry area should be as long as the longest information.
- ___ R 8 If window space is limited, an entry area may contain scroll bars.
- ___ R 9 The entry area should include a decimal point if the entry is in decimal format.
- ___ R 10 The format of each text entry field should be consistent with users' expectations.
- ___ R 11 Whenever possible, the entry for a field should be in "chunks" that are meaningful to users.
- ___ R 12 Strings over 5-7 characters long should be entered in 3-4 character units plus a delimiter.
- ___ R 13 Data should be automatically entered in a field when it is known or can be computed.

Section 5.4.2 Distinguishing Text Entry Areas from Noneditable Text

- ___ M 1 Text that can be edited shall be displayed in a text entry field.
- ___ R 2 Noneditable text should be presented directly and not within a box.
- ___ R 3 If noneditable text must be displayed in a box, the box should have a different appearance.
- ___ R 4 If the pointer is in an information-only area, the pointer shape should not change to an I-beam.
- ___ R 5 Clicking in an information-only area should not change the area or produce a text cursor.

Section 5.4.3 Text Entry in a Text Entry Field

- ___ R 1 With multiple text entry fields, the text cursor should appear in the upper leftmost field.
- ___ R 2 If the field contains text, the text should be highlighted.
- ___ R 3 Users should perform text entry as described in Section 2.3.
- ___ R 4 Editing commands should be available as appropriate for the task being performed.
- ___ R 5 For variable-length text entry, information should be automatically justified or truncated.
- ___ R 6 Users should be able to enter numeric data from the keyboard or the numeric keypad.
- ___ R 7 The amount of data users have to enter in a text entry field should be minimized.
- ___ R 8 Users should be able to enter abbreviations and not have to enter units of measurement.

Section 5.4.4 Moving between Text Entry Fields

- ___ R 1 The text cursor should be moved between areas with the Tab or arrow keys, not done automatically.
- ___ R 2 When moved to another text entry area, the text cursor should appear at the left of the area.
- ___ R 3 If the text entry area already contains text, the text should highlight.
- ___ R 4 The text cursor should be placed by pointing to any text area and pressing the L button.
- ___ R 5 The size of the text cursor should change to match the size of text font where it is placed.

Section 5.5 List Boxes

- ___ M 1 A list shall be used to present a set of items from which to choose.
- ___ M 2 The items in the list shall be displayed vertically, with one item per line.
- ___ M 3 The title of the list shall describe its purpose or contents and appear above the box.
- ___ M 4 A vertical scroll bar shall appear to the right of the items in the list box.
- ___ M 5 Users shall be able to move the slider in the scroll bar when the list exceeds the space available.
- ___ R 6 The list box should scroll only in response to user action and not scroll automatically.

Version 1.1, Change 1

- ___ R 7 A list box should be wide enough to read the items without having to scroll horizontally.

Section 5.5.1 Ordering Items in a List Box

- ___ R 1 The items in a list should be presented in a natural or expected order.

Section 5.5.2 Adding Items to a List Box

- ___ R 1 Items added to a list box should appear in their correct position in the list.

Section 5.5.3 Searching Items in a List Box

- ___ M 1 Users shall be able to search a list by moving the slider until the item appears.
- ___ R 2 Users should be able to perform a speed search or an incremental search.
- ___ R 3 For speed search, type first letter of item; list auto scrolls to first instance with that letter.
- ___ R 4 For incremental search, type the first few letters and press Return; list auto scrolls to first match.
- ___ R 5 Neither speed search nor incremental search should be case-sensitive.
- ___ R 6 If the search must be case-sensitive, then this information should be provided to users.

Section 5.5.4 Selecting Items in a List Box

- ___ M 1 To select an item in a list, users shall point to the item and click the L button.
- ___ M 2 With a default push button, users shall double click on a list item to execute the default.
- ___ M 3 Users shall follow standard methods (table 2-2) to select multiple items in a list.
- ___ R 4 If a text entry field is included with a list, the field should behave like an incremental search.
- ___ R 5 Applications should not use a selection box in the manner recommended by the Motif Style Guide.

Section 5.5.5 Sorting Items in a Multi-Column List Box

- ___ R 1 A standard approach should be provided for sorting records in a multi-column list box.
- ___ R 2 Users should be able to sort the records by selecting the column heading.
- ___ R 3a The headings of columns that can be sorted should appear as push buttons.
- ___ R 3b When users click on the heading, the records in the matrix are sorted in ascending order.
- ___ R 3c The button for the heading remains highlighted to indicate the column that was sorted.

Section 5.6 Scroll Bars

- ___ M 1 Scroll bars shall be used to view textual/graphic information when it exceeds space available.
- ___ M 2 Users shall be able to scroll to the top or the bottom of the information but not beyond.
- ___ M 3 Clicking on a stepper arrow shall make arrow highlight and move one unit in the arrow direction.
- ___ M 4 To move continuously through the information, users shall press on the arrow.
- ___ M 5 Clicking on the scroll region shall make the window move one window minus one unit at a time.
- ___ M 6 To move the window continuously, users shall press on the scroll region.
- ___ M 7 Dragging the slider shall make the slider move; window shall move consistent with slider location.
- ___ R 8 Pointing to a scrollable area and dragging should make it scroll in the pointer direction.
- ___ R 9 The information should scroll at the same speed as when users press on a stepper arrow.
- ___ R 10 Scrolling should stop if pointer moved into or outside window or L button released.

Section 5.7 Scales

- ___ M 1 A scale shall be used to set or display a value in a range.
- ___ M 2 Users shall display a value by adjusting a slider to a specified position along a line.
- ___ M 3 A scale shall have scale bar/marks, a slider, and a label showing the slider value.
- ___ R 4 The scale should be labeled with the minimum and maximum values for the scale.
- ___ M 5a To display a value, users shall point at the value on the scale and click the L button.
- ___ M 5b The slider shall move to location of value; its value shall be displayed above or alongside.
- ___ M 6a To display a value, users shall point to the slider, press the L button, and drag the slider.
- ___ M 6b The value displayed in the label shall change as users drag the slider.

Version 1.1, Change 1

- ___ M 6c When the value displayed is the value desired, users shall release the L button.

SECTION 6. SYSTEM-LEVEL WINDOWS

Section 6.1.1 Login Procedure

- ___ M 1 Each system shall implement a login procedure to be completed before system functions are accessed.
- ___ M 2 A system shall make available only applications for which the user is allowed access.
- ___ R 3 If system is not available, a message should indicate system status and when available.

Section 6.1.2 The Login Window

- ___ M 1 A login window shall appear on the screen when users begin a session on a system.
- ___ M 2 The login window shall contain two text entry fields for user identification and password.
- ___ M 3 The text entry areas shall be longer than needed to not give clue to the number of characters required.
- ___ M 4 The appearance and behavior of the objects in the login window shall be consistent with table 7-1.
- ___ M 5 Users must enter a valid identification and password before a session is initiated.
- ___ M 6 If users enter an invalid identification or password, an error message shall appear in the window.
- ___ M 7 Users who fail repeatedly to log on shall be locked out and told to contact the administrator.

Section 6.1.3 Feedback during System Start-Up

- ___ R 1 During start-up, "unavailable" message displayed, the pointer is an hourglass, and input is disabled.
- ___ R 2 When ready, the message disappears, the pointer is standard shape, and input is enabled.
- ___ R 3 If appropriate, messages should display status, e.g., response time and known unavailability.

Section 6.2.1 Window Design

- ___ M 1 The system window shall appear on the screen when system start-up is complete.
- ___ M 2 The system window shall cover the entire root window to prevent accessing the operating system.
- ___ M 3 The system window shall contain classification and menu bars at the top and an optional title.
- ___ M 4 The current classification level shall appear in the middle of the classification bar.
- ___ M 5 A digital clock may be displayed at the right of the classification bar, and status at the left.
- ___ M 6 The system menu bar shall list the titles of the menus available at the system level.
- ___ M 7 These menus shall provide access to the set of application programs available within the system.
- ___ M 8 The remainder of the system window shall be available for displaying application windows.
- ___ M 9 Action icons may be used to provide access to frequently executed actions within a system.
- ___ R 10 Action icons common to all applications should be placed along the left margin (from the top down).
- ___ R 11 Application-level icons should be displayed at the lower left corner of the screen.

Section 6.2.2 Window Appearance and Behavior

- ___ R 1 The system window should have the appearance and behavior per table 6-1 and figure 6-3.
- ___ M 2 System window shall not move/resize; classification and system menu bars shall not be obscured.
- ___ M 3 System window shall always be active so that users can select help or system-level menu options.
- ___ M 4 Each system shall specify the initial appearance of the system window following start-up.
- ___ M 5 A default application may open at start-up, with a primary window appearing automatically.
- ___ M 6 The system may select a default application, based on the type of user.

Section 6.2.3 Classification Markings

- ___ M 1 The system shall control the classification marking and show the current highest level of data.
- ___ M 3 The color for Unclassified is green (ForestGreen; 34 139 34).
- ___ M 4 The color for Confidential is blue (CornflowerBlue; 100 149 237).
- ___ M 5 The color for Secret is red (Red; 255 0 0).
- ___ M 6 The color for Top Secret is orange (DarkOrange; 255 140 0).
- ___ M 7 These colors shall be used as the background in the classification portion of the classification bar.

Version 1.1, Change 1

Section 6.5.1 User-Initiated Logout

- ___ R 1 To end a session, users should select a Logout option from the System menu.
- ___ M 2 Users shall be prompted to confirm Logout if there are any unsaved data.
- ___ R 3 Where appropriate, users should be prompted to log out of any applications.
- ___ M 4 During logout, all processing in application windows shall stop, and all windows shall be removed.
- ___ M 5 When logout is complete, the initial login window shall be displayed.

Section 6.5.2 Automatic Logout

- ___ R 1 If there is auto logout, a standard amount of user inactivity should be defined before logout occurs.
- ___ R 2 The time period before auto logout should be modifiable by users.
- ___ R 3 A message should be displayed during inactivity indicating action needed to avoid auto logout.
- ___ R 4 An auditory signal should be presented at intervals during the period of inactivity.
- ___ R 5 For auto logout, unsaved data should be saved with a message indicating logout and the file name.

SECTION 7. APPLICATION-LEVEL WINDOWS

Section 7.1.1 Primary and Dialog Windows

- ___ M 1 All primary and dialog windows shall contain a title and a main area.
- ___ 2 Including a window menu bar and a message bar depends on window type window and functionality.

Section 7.1.1.1 Window Title

- ___ M 1 A window title shall appear in the title bar of the window.
- ___ M 2 The title shall be centered in the title bar and presented in upper-case letters.
- ___ R 3 Each window title in an application should be unique.
- ___ R 4 If an application has multiple primary windows, each title should indicate the window purpose.
- ___ R 5 The title of a dialog window should describe its purpose.
- ___ R 6 If a dialog window is selected from a pull-down menu, the title should match the option wording.

Section 7.1.1.2 Window Menu Bar

- ___ M 1 If a window includes a menu bar, it shall appear below the title bar.
- ___ R 2 The menu bar should contain no more than ten menu titles plus Help.
- ___ R 3 The titles should begin at the left margin and extend rightward, with Help at the right margin.
- ___ R 4 Options in window menus should be unique and not duplicate options in system menus.
- ___ R 5 Titles of window menus should be unique and not duplicate the titles of system menus.
- ___ R 6 If the same menu is in different menu bars, the menu should always have the same title and actions.

Section 7.1.1.3.1 Arrangement of Window Controls

- ___ R 1 The main area of a window should be organized by the information presented or the action required.
- ___ R 2 Controls that perform similar or related functions should be presented together in a control panel.
- ___ R 3 The panel should have a simple frame and be clearly labeled to indicate the function performed.
- ___ R 4 The controls should be grouped into rows or columns as appropriate to the function.
- ___ R 5 Controls for similar options should be together, and different options should be grouped separately.
- ___ M 6 If scroll bars are needed, they shall be located to the right or at the bottom of the area.
- ___ R 7 If certain controls become unavailable, they should be dimmed and not available for selection.
- ___ R 8 If certain controls are never available to users, they should not appear in the window.

Section 7.1.1.3.2 Arrangement of Push Buttons

- ___ R 1 The bottom of the main area should be reserved for push buttons indicating actions that can be taken.
- ___ R 2 The push buttons should be displayed in a row that is centered at the bottom of the window .
- ___ R 3 Push buttons should be ordered from L-R in sequence of use, with most frequently used on the left.

Version 1.1, Change 1

- ___ M 8 Classification marking shall be upper-case letters, in at least 14-pt bold font.
- ___ R 9 Each system should provide an option for implementing alternate classification color sets.

Section 6.3.1 Integration of Applications into System-Level Menus

- ___ 1 Application functions may be available as a full menu, a group of options, or a single option.
- ___ 2 Application functions may be available from the system menu or from within another application.

Section 6.3.2 Organization of the System Menu Bar

- ___ R 1 The set of menu titles that appear in the system menu should describe the overall functionality.
- ___ R 2 The menu bar should contain no more than ten menu titles plus Help.
- ___ R 3 Menu titles should begin at the left margin and extend rightward and be easy to read.
- ___ R 4 The leftmost menu titles should provide functions related to map and situation displays.
- ___ R 5 The middle menu titles should include database management, msg handling, and decision support.
- ___ R 6 The rightmost menu titles should provide system-related functions.
- ___ R 7 Help should appear at the far right margin of the menu bar.

Section 6.3.3 Consistency in Menu Titles between Systems

- ___ R 1 Each system should determine when it provides the same functionality as other Navy C2 systems.
- ___ R 2 The same system-level menu titles should be used when accessing the same functionality.
- ___ R 3 An application in multiple systems should have the same name when it appears in a system menu.

Section 6.3.4 Access to Menu Options within an Application

- ___ R 1 If a menu option is available from within an application, it should be displayed in normal text.
- ___ R 2 Options that are inappropriate to execute from within the application should be dimmed.
- ___ R 3 If all of the options in a menu are inappropriate, the menu title should be dimmed.
- ___ R 5 If only one instance of an application can run, the menu option should be dimmed during interaction.
- ___ R 6 If multiple instances can run, the menu option should return to normal to indicate it can be selected.
- ___ R 7 When working in an application, users should be able to select Help from the system menu bar.
- ___ R 8 Users should be able to browse all titles in the system menu bar even if all are unavailable.

Section 6.3.5 Quick Access to System Functions

- ___ R 1 Each system should include navigation aids as an alternative to the function-oriented menus.
- ___ R 2 These aids should be accessible through Help in the system window.

Section 6.4.1 System Support Functions

- ___ R 1 A system should include the resources and utilities needed to support overall tactical functionality.
- ___ R 2 Support functions should be provided in the Utilities and System menus in the system menu bar.
- ___ R 3 The Utilities menu should include basic utilities, COTS software, macros, and briefing support.
- ___ R 4 The System menu should include logout, print, status, preferences, alerts, peripherals, and files.
- ___ R 5 Users should be able to print the entire screen or a window with/without secondary windows.
- ___ R 6 Printing should include the classification marking and produce an accurate color shading.
- ___ R 7 Users should be able to work with files without detailed knowledge of the file structure.
- ___ R 8 Each system should implement a screen saver and a means to rapidly suppress a display .
- ___ R 9 Users should be allowed to temporarily suspend a session without logging out.
- ___ R 10 During suspension, processing should continue, but allow no interaction until user resumes session.

Section 6.4.2 User-Specified Interface Settings

- ___ R 1 A system should provide user-specified settings that support overall system functionality.
- ___ R 2 A default value for each parameter should be set for all applications, unless changed by users.
- ___ R 3 Users should be able to review the parameters and reset them at any time during a session.
- ___ R 4 When users end the session, session-specific settings should revert to the default values.

Version 1.1, Change 1

- ___ R 4 Buttons for positive actions should be on the left, followed by negative and cancelling actions.
- ___ R 5 If a default push button is included in a window, it should be the leftmost button.
- ___ R 6 A Help button should be included in every window and should be the rightmost button.
- ___ R 7 Push buttons should appear in the same order throughout the application.
- ___ R 8 Push button order in modal windows: OK, Cancel, Help.
- ___ R 9 Order in modeless windows: OK, Apply, Cancel, Help, or OK, Apply, Reset, Cancel, Help.
- ___ R 10 The OK button should be the default in modal windows and in modeless windows that are transient.
- ___ R 11 Apply should be the default in modeless windows that are displayed for multiple actions.
- ___ R 12 If their actions affect different objects/data, push button labels should reflect what each affects.
- ___ R 13 Push buttons should be arranged near the object(s) to which they are related.
- ___ R 14 Push buttons related to overall window functionality (e.g., OK, Cancel) should be at the bottom.

Section 7.1.1.3.3 Arrangement of Action Icons

- ___ R 1 Action icons should be bound to the application, i.e., not movable outside the application window.
- ___ R 2 There should be no more than 20 action icon buttons; they should appear along the left margin.

Section 7.1.1.4 Message Bar

- ___ R 1 A message bar should appear at the bottom window margin for noncritical application messages.
- ___ R 2 The left side of the message bar should be for routine messages, simple help, and window status.
- ___ R 3 The right side of the message bar should be used to identify information about the window.
- ___ R 4 A message bar should be part of the primary window but may also be included in a dialog window.
- ___ R 5 The message bar may be an area that is set aside for this purpose or may be a temporary area.

Section 7.1.2 Message Windows

- ___ R 1 A message window should be as small as possible, yet large enough for the information.
- ___ R 2 A message window should have a unique appearance and appear at a standard location.
- ___ R 3 A message window should include title, symbol for the type, message, and one or more push buttons.
- ___ R 4 The title of a message window should indicate the process/application generating the message.
- ___ R 5 Users should be able to identify the source of the message (e.g., an application that is iconified).
- ___ R 6 The message itself should be meaningful and require no documentation or translation.
- ___ R 7 Error messages should focus on the procedure for correcting the error, not the cause of the error.
- ___ R 8 For a repeated error, the messages should include recommendations for preventive actions.
- ___ R 9 A message window should include at least one push button along the bottom of the window.

Section 7.1.2.1 Progress Message Windows

- ___ R 1 Applications should display a progress message window for processing time over 5 sec.
- ___ R 2 The progress window should have a working symbol, message, and OK, Cancel, Help push buttons.
- ___ R 3 For lengthy processing time, the window should update to indicate the status of processing.
- ___ R 4 A progress window should not interrupt processing by the application.
- ___ R 5 The window should remain displayed until action is complete or window doing action is iconified.
- ___ R 6 When processing is complete, the progress window should disappear.
- ___ R 7 Users should be able to cancel an operation and should confirm this if unsaved data will be lost.

Section 7.1.2.2 Warning Message Windows

- ___ R 2 Applications should display a warning message window for critical consequences of an action.
- ___ R 3 A warning window should be displayed for attempts to execute a destructive action.
- ___ R 4 The window should have symbol, message, and Yes, No, Help, or OK, Cancel, Help push buttons.
- ___ R 5 When a warning window is displayed, processing should be suspended until users decide on action.

Section 7.1.2.3 Error Message Windows

- ___ R 1 An error message window should be displayed when an error occurs.

Version 1.1, Change 1

- ___ R 2 The window should have symbol, message, OK, Cancel, Help, or Retry, Cancel, Help push buttons.
- ___ R 3 The application should suspend processing until users respond to the window.
- ___ R 4 If processing can be reinstated, the window should include a Retry push button.

Section 7.1.2.4 Information Message Windows

- ___ R 1 Information message windows should convey noncritical information requiring acknowledgement.
- ___ R 3 The window should have symbol, message, and OK, Help push buttons.
- ___ R 4 An information window should not interrupt processing and should remain until users respond to it.
- ___ R 5 Timed-information windows should not be displayed and then resume processing without a response.

Section 7.1.2.5 Question Message Windows

- ___ R 1 A question message window should request clarification of a previous response.
- ___ R 2 The window should have symbol, message, Yes, No, Help, or Yes, No, Cancel, Help push buttons.
- ___ R 3 The application should suspend processing until users respond to the window.

Section 7.2.1 Color of Objects

- ___ M 1 Interface objects shall have the appearance indicated in table 7-1 and in figure 7-10.
- ___ M 2 The shadow colors for each object shall be those that are generated automatically by Motif.
- ___ R 3 Applications modifying basic color appearance should do so per section 7.4.

Section 7.2.2 Size of Objects

- ___ R 1 Menus should be wide enough for the wording of the longest option plus a keyboard accelerator.
- ___ R 2 Push buttons should be the same size within a window.
- ___ R 3 If push buttons contain text labels, the buttons should be wide enough for the longest button label.
- ___ R 4 If the push buttons contain icons, the buttons should be large enough to display the largest icon.
- ___ R 5 Radio and check buttons should be the same size whenever they appear in a window.
- ___ R 6 Text entry fields and list boxes should be wide enough to read all of the text being presented.
- ___ R 7 If necessary, horizontal and vertical scroll bars should be used to view text and list contents.
- ___ R 8 Enough space should be provided between menus so that each can be easily read by users.

Section 7.2.3 Consistent Appearance and Behavior

- ___ M 1 Objects shall be identifiable solely by appearance; the same type shall have the same appearance.
- ___ M 2 An object in an application shall appear and behave as indicated in these interface specifications.
- ___ M 3 A new object or existing object shall not be used in a way that conflicts with these specifications.
- ___ R 4 Users should be able to distinguish objects that are similar in shape by distinctive visual cues.
- ___ R 5 The appearance and behavior of a new object should be consistent with these specifications.
- ___ R 6 New objects may be created for functionality not in these specifications.

Section 7.3.1 Text Font, Size, and Readability

- ___ M 1 Normal, dimmed, and highlighted text shall appear and behave as indicated in table 7-1.
- ___ R 2 Black text should be used; any different text color should promote good readability.
- ___ R 3 Text should be of sufficient size and thickness to be readable at normal viewing distance.
- ___ R 4 Minimum character height should be 1/200 of viewing distance (e.g., at least 14-point type).
- ___ R 5 Maximum text size should not exceed 10 percent of the available vertical display area.
- ___ R 6 Text for a briefing presentation should be readable at the normal audience viewing distance.
- ___ R 7 User-selectable font size should be provided when a single font does not satisfy these requirements.

Section 7.3.2 Capitalization

- ___ R 1 Titles and major headings should be presented in upper-case letters.
- ___ R 2 All other text should use a combination of upper- and lower-case, following standard rules.
- ___ R 3 A consistent approach to capitalization should be used to minimize textual distraction.

Version 1.1, Change 1

- ___ R 4 All upper-case letters should be used only for acronyms and abbreviations and for emphasis.
- ___ R 5 Arabic rather than Roman numerals should be used when information has to be numbered.

Section 7.3.3 Acronyms and Abbreviations

- ___ R 1 Acronyms and abbreviations should be shorter than full names and commonly understood.
- ___ R 2 Abbreviations should be the shortest possible that will ensure uniqueness.
- ___ R 3 Abbreviations should be meaningful, recognizable, and used consistently.
- ___ R 4 Words not commonly abbreviated should not be abbreviated.
- ___ R 5 Acronyms and abbreviations should comply with MIL-STD-12, MIL-STD-411, and MIL-STD-783.
- ___ R 6 New acronyms should be generated according to rules contained in MIL-STD-12.
- ___ R 7 When abbreviations or acronyms are used, a dictionary should be available to users (e.g., in Help).

Section 7.3.4 Noneditable Text

- ___ R 1 Consistent grammatical structure should be used for all noneditable text.
- ___ R 2 Wording should be consistent and use familiar terms and task-oriented language of users.
- ___ R 3 Blocks of text should be broken into smaller, meaningful groups.
- ___ R 4 Continuous text should be phrased in simple sentences, in the affirmative, and in active voice.
- ___ R 5 A sequence of events or steps should be presented in the order they are performed.
- ___ R 6 The referent for "it" or "they" in a sentence should be easily identified.
- ___ R 7 Normal punctuation rules should be followed, and contractions and hyphenation should be avoided.

Section 7.3.5 Editable Text

- ___ R 1 The manner in which editable text is displayed should be appropriate to the task being performed.
- ___ R 2 Stored text should be shown in standard format; text editing should be converted into this format.

Section 7.4 Color Use in Windows

- ___ R 1 Color (beyond table 7-1) should be used only to highlight, group, relate, or add meaning.
- ___ R 2 Applications should select colors to be consistent with the specifications contained in table 7-1.
- ___ R 3 Color pairs at spectral extremes should not be used together because they appear to vibrate.
- ___ R 4 White text should not be displayed on a black background because this makes the text less readable.
- ___ R 5 Blue should not be used as a text color or for any critical information.
- ___ R 6 Color should be used consistently and in ways that match user expectations.
- ___ R 7 Red should be used to indicate critical or irreversible options.
- ___ R 8 Noncritical options should not be colored red.
- ___ R 9 Colors for status indicators should be the same throughout and be restricted to that function.
- ___ R 10 Color in message windows should appear in the symbol and not used for text.
- ___ R 11 Red should only alert that an application/process is inoperative, possibly until corrective action.
- ___ R 12 Yellow should only alert to situations involving caution, recheck, or unexpected delay.
- ___ R 13 If color is used to impart meaning, it should be used as a redundant code.
- ___ R 14 If color coding is used, each color should represent one category of displayed data.
- ___ R 15 The number of colors for alphanumeric display should not exceed 7, with only 4 codes at a time.
- ___ R 16 The number of colors for graphical displays should not exceed 8-9 for good discrimination.
- ___ R 17 Slight shade changes in color should not be used to show gradation or choice.
- ___ R 18 The background color behind text should not be changed to show a change in system status.
- ___ R 19 Change in system status should be signaled by changing the color of an object next to the text.
- ___ R 20 A black background should not be used for color-coded items because black has poor contrast.
- ___ R 21 Unsaturated hues (e.g., tan, offwhite) are good background colors; pure white is too bright.

Section 7.5.1 Selecting Objects to Match User Actions

- ___ R 1 Objects should be selected to match the actions that users will execute in the window.
- ___ R 2 Use primary window(s) for the primary actions and to present frequently used controls.

Version 1.1, Change 1

- ___ R 3 Use dialog windows for ancillary application actions and to present less-frequently used controls.
- ___ R 4 Use radio buttons for selecting one option from up to six mutually exclusive options.
- ___ R 5 Use an option menu when more options (no more than 10-12) are available.
- ___ R 6 Use an option menu rather than radio buttons when space is limited.
- ___ R 7 Use a list box when users need to make a choice from a large group of options (more than 12).
- ___ R 8 Use radio buttons or an option menu when the set of options is not likely to change.
- ___ R 9 Use a list box when the options may change.
- ___ R 10 Use check buttons to choose from up to 7 items in a group; use a list box otherwise.
- ___ R 11 Use push buttons for frequently used selections or when the pointing device is already being moved.
- ___ R 12 Use pop-up menus only to access functions without moving the pointing device.
- ___ R 13 Do not use pop-up menus for frequently executed operations because they require user recall.
- ___ R 14 Use option menus for setting values or choosing from a set of related items.

Section 7.5.2 Arranging Information to Match User Actions

- ___ R 1 For displaying pageful of information, push buttons should be provided to perform standard paging.
- ___ R 2 Window layout should support natural scanning order and probable selection sequences.
- ___ R 3 In most cases, scanning order in windows should be from left to right and top to bottom.
- ___ R 4 The most frequently used radio or check button should be the top (or leftmost) button.
- ___ R 5 Window layout should be designed to be logical to users and appropriate to user actions.

Section 7.5.3 Arranging Information by Importance

- ___ R 6 The most important information and controls should be in the upper left part of the window.
- ___ R 7 The objects in a window should be arranged to accommodate users resizing the window.
- ___ R 8 Task-critical information should be visually set apart from other information.
- ___ R 9 There should be at least 1 space above/below and 2 spaces before/after critical information.

Section 7.5.4 Designing Windows to Minimize User Memory Load

- ___ R 1 Windows should be designed to minimize the short-term memory load placed on users.
- ___ R 2 A window should contain all of the information relevant to the task.
- ___ R 3 User should be able to complete a task without referring to external information.

Section 7.5.5 Coding Critical Information in Windows

- ___ R 1 Capitalization should not be used as the sole indication of critical information in a window.
- ___ R 2 Attention should be focused on critical information by bolding/ brightening, color coding, etc.
- ___ R 3 Color and sound should be used for critical messages, and a response required before terminating.
- ___ M 4 Auditory signals shall be used to alert to critical conditions or operations.
- ___ R 5 For noncritical auditory alarms, a simple action should acknowledge/turn off the signal.
- ___ R 6 Auditory signals should be intermittent in nature and allow sufficient time to respond.
- ___ R 7 Auditory signals should be distinctive in intensity and pitch and not exceed four.
- ___ R 8 Signal intensity, duration, and location should be appropriate to environment and personnel.
- ___ R 9 Flash coding should be used only to display urgent information for user attention.
- ___ R 10 No more than two levels of flash coding should be used.
- ___ R 11 Flash rate should be 3-5 Hz with equal on/off; for 2 levels, the second should be less than 2 Hz.
- ___ R 12 For flash coding of displayed item, a flashing symbol should be used rather than blinking the item.
- ___ R 13 Users should be able to acknowledge the event causing the flashing and suppress it if desired.

Section 7.5.6 Dynamic Information in Windows

- ___ R 1 For dynamically changing information, users should be able to control the rate of update.
- ___ R 2 Users should be able to freeze an updated display, then resume at the stoppage or the current time.
- ___ R 3 When reading dynamically changing information, update values no more than once per second.
- ___ R 4 When identifying rate of change or reading gross values, update values 2-5 times per second.

Version 1.1, Change 1

- ___ R 5 Users should be alerted to critical information in an inactive or iconified window.
- ___ R 6 For a frozen dynamic window, feedback should prompt users to return to automatic updating.
- ___ R 7 Users should be informed if significant changes in data occurred while the display was frozen.

Section 7.5.7 Consistency in Design Across Windows

- ___ R 1 A consistent organizational scheme should be adopted for the key elements in windows.
- ___ R 2 The same window design should be employed whenever users have to perform the same basic task.
- ___ R 3 A single design should be used to present identifying information on all data records.
- ___ R 4 Different or distinctive elements can be used to fit a task, if consistent across windows.

Section 7.6.1 Initial Window Appearance

- ___ R 1 Developers should determine the appearance for each window when it first appears on the screen.
- ___ R 2 Developers should determine what information should be visible when a window appears.
- ___ R 3 A scrollable window should display the topmost information and show a complete line of text.
- ___ R 4 Developers should determine the appropriate settings for the objects in a window when it appears.
- ___ R 5 When button(s) are selected but not executed, the selection should not be saved.

Section 7.6.2 Initial Window Size and Placement

- ___ R 1 An application window that covers all of the screen should allow users to resize it.
- ___ R 2 Each window should be sized so that all objects in the window are visible when it first appears.
- ___ R 3 A window should be wide enough to read the title and tall enough to read the title and menu bar.
- ___ R 4 Developers should define the initial placement for each window when it first appears on the screen.

Section 7.6.3 Resizing

- ___ R 1 When a window is resized, only the window borders and not the size of objects should change.
- ___ R 2 When a text window is enlarged, it should show more lines of text but text size should not change.
- ___ R 3 The contents of the window should remain visible during resizing so that users can view the effect.
- ___ R 4 Horizontal and vertical scroll bars should appear as appropriate when a window is resized.
- ___ R 5 Window resizing should be limited so that objects are not obscured at minimum size.
- ___ R 6 With multiple panes, developers determine how the space is divided between panes after resizing.
- ___ R 7 Depending on functionality, pane size/contents may change automatically during resizing.

Section 7.6.4 Order of Window Interaction

- ___ R 1 For each dialog window, developers specify the windows with which users are allowed to interact.
- ___ R 2 Developers specify if modeless, primary modal, application modal, system modal interaction.

Section 7.6.5 Processing in Iconified Windows

- ___ R 1 Users should be kept informed when processing is ongoing in an iconified window.
- ___ R 2 Users should be informed whenever critical processing events occur in an iconified window.

SECTION 8. TASK-SPECIFIC WINDOWS

Section 8.1.1 System-Level and Application-Level Help

- ___ R 1 Help should be available to users both at the system level and within individual applications.
- ___ R 2 System help should describe capabilities and may have tutorial and navigation aids.
- ___ R 3 Application help should include general information on functionality and may include index.
- ___ R 4 Help push button in a dialog window should provide information on actions in the window.
- ___ R 5 Help message in the message bar of a window should explain how to complete an action.
- ___ R 6 Applications can provide access to help via an action icon rather than a menu title.

Section 8.1.2 Help Window Design

Version 1.1, Change 1

- ___ R 1 A help window should include a title, a main scrollable area, and an OK push button at the bottom.
- ___ R 2 The push button in a help window should appear and behave as the default action.
- ___ R 3 When a help window appears, it should be wide enough to read the text.
- ___ R 4 When a help window appears, it should display information at beginning of help description.
- ___ R 5 When a help window appears, it should not obscure what it is describing.
- ___ R 6 Users should be able to move and resize a help window.
- ___ R 7 Users should be able to keep the help window on the screen while working in another window.
- ___ R 8 Users should be able to print any help window by selecting all or marking beginning/end of part.
- ___ R 9 When done with help, users should have an easy means to return to where they were working.

Section 8.1.3 Help Window Content

- ___ R 1 Help information should explain the steps required to execute the action(s) in the window.
- ___ R 2 Help information should be limited to the actions that can be performed in the window.
- ___ R 3 Help information should be worded concisely, using consistent structure, phrasing, and punctuation.
- ___ R 4 When presenting a sequence of steps, the explanation should follow the same sequence.
- ___ R 5 Help information should assume users have had previous training and only need reminding.
- ___ R 6 Users should be able to easily scan help text; steps should be numbered and in separate paragraphs.
- ___ R 7 Help should decode abbreviations/acronyms and explain behavior of user-selectable objects.
- ___ R 8 A help window should be removed when the related window is iconified or closed.

Section 8.1.4 Online Training

- ___ R 1 Online training should be available at system and application levels as a separate module.
- ___ R 2 Online training should be designed to accommodate different levels of users.
- ___ R 3 Online training should use simulated data and protect real system data from change/corruption.
- ___ R 4 The training mode should be clearly indicated to users and kept separate from normal system use.

Section 8.2.1 Data Entry Window Design

- ___ R 1 A data entry window should include a title that describes the contents of the window.
- ___ R 2 If multiple pages, each page should have same title, current page, and total pages indicator.
- ___ R 3 If possible, related data fields should appear on the same page.
- ___ R 4 The numbering of items should be continuous from one page to the next.
- ___ R 5 The design of each data entry window should be consistent with the task being performed by users.
- ___ R 6 Data fields should be organized by sequence of use, frequency of use, or importance.
- ___ R 7 Data fields should be organized with related fields together and unrelated fields separated.
- ___ R 8 When users work from a hardcopy form, the window format should have an identical format.
- ___ R 9 For tabular data, entry areas should be arranged in rows/columns, with each row/column labeled.
- ___ R 10 If multiple pages, the row and column labels should remain visible along the page edges.
- ___ R 11 Every fifth row of data should be separated by a blank line or other delimiter.
- ___ R 12 Information should be automatically justified; numeric data aligned at the decimal point.
- ___ R 13 A data entry window should contain the controls needed to support data entry and manipulation.
- ___ R 14 If a data entry window has more than one page, the window should include paging controls.
- ___ R 15 If users have to make multiple entries, the window should include Clear and Restart controls.
- ___ R 16 If push buttons are used in a data entry window, they should be separate and at the bottom.
- ___ R 17 A data entry window should allow users to obtain information about a data field and its contents.

Section 8.2.2 Data Field Format

- ___ R 1 Data field names should be unique; the same names should be applied throughout the application.
- ___ R 2 Data fields should be identified as mandatory or optional; optional fields should be labeled as such.
- ___ R 3 Data fields should be text entry fields (users type) or option menus (users select).
- ___ R 4 Data fields with values that cannot be changed should be in an information-only area.

Version 1.1, Change 1

Section 8.2.3 Data Entry and Manipulation

- ___ R 1 The same set of actions to enter and manipulate data should be used throughout the application.
- ___ R 2 Text cursor should be placed using the pointing device and moved with keyboard/pointing device.
- ___ R 3 Do text entry per section 5.4.2; move per section 5.4.3; make selections per section 4.3.
- ___ R 4 Users should be able to tab to adjacent fields, and tabbing always moves in the same direction.
- ___ R 5 Users should be able to back up to any field and change it prior to entering data into the system.
- ___ R 6 A validity check should be done on the data entered, with a visual and/or auditory error message.
- ___ R 7 Data should be entered with an explicit action, e.g., Enter menu item, Apply or OK push button.
- ___ R 8 If appropriate, the application should provide feedback to users to indicate successful data entry.
- ___ R 9 Users should be able to enter data into the system at any time during data entry.

Section 8.2.4 Data Query

- ___ R 1 Data query language should reflect the perceived structure of the data.
- ___ R 2 The language should allow users to specify the data to retrieve, then to display and manipulate it.
- ___ R 3 Users should be able to request data without having to tell the system how to find it.
- ___ R 4 Queries should use operationally meaningful terminology and not reflect how the data are stored.
- ___ R 5 Users should be able to construct simple/complex, predefined queries and save/retrieve/execute.
- ___ R 6 The language should permit alternate forms of the same query using natural language.
- ___ R 7 The application should prompt users to confirm a query if data retrieval time will be excessive.

Section 8.3.1 Text Window Design

- ___ R 1 A text window should be wide enough to display an entire line of text without scrolling.
- ___ R 2 A text window should include vertical scroll bars and indicator of current location in the document.
- ___ R 3 Users should be able to save, access, retrieve, rename, and print the document.
- ___ R 4 Users should be able to specify the portions of the document to be printed and the printer.
- ___ R 5 The system should acknowledge the print command and provide the status of the printer and queue.

Section 8.3.2 Text Manipulation

- ___ R 1 Users should be able to easily specify the format of a document and the font type, size, and style.
- ___ R 2 Automatic line break and wordwrap at the right margin should be available.
- ___ R 3 Automatic pagination (page numbers based on the number entered by users) should be available.
- ___ R 4 The window should be formatted as the printed output, or there should be option to see this format.
- ___ R 5 A copy of the original document should be retained until users confirm that it is to be changed.
- ___ R 6 The document should not be modified automatically as users make each editing change.
- ___ R 7 Applications should provide both search and search/replace capabilities for users.
- ___ R 8 For search, users should type the text string (not case-sensitive); the first instance is highlighted.
- ___ R 9 For search/replace, users type text string to be searched and change desired (is case-sensitive).

Section 8.4.1 Tabular Data Window Design

- ___ R 1 Tabular data windows should display alphanumeric information in tabular or columnar form.
- ___ R 2 Tabular data windows should be used for information display only.
- ___ R 3 Single- or multi-columned list boxes should be provided for users to select one or more items.
- ___ R 4 The window should include a title and vertical/horizontal scroll bars if space is exceeded.
- ___ R 5 If the data cover several pages, pages should be numbered and include push buttons for paging.
- ___ R 6 Column labels should be identical for all pages; the last line on a page is first line on next page.
- ___ R 7 The content of the window should not extend over more than one page horizontally.
- ___ R 8 A database query output window should include the query that was executed.

Section 8.4.2 Tabular Data Window Content

- ___ R 1 Each tabular data column should have a heading; columns should be separated by at least 3 spaces.
- ___ R 2 Data groupings should be indicated with blank space, separator lines, and/or different intensity.

Version 1.1, Change 1

- ___ R 3 Multiple colors should be used only if the colors provide additional meaning.
- ___ R 4 A blank space should be inserted after every third to fifth row and column of tabular data.
- ___ M 5 Left-justify alphabetic data; right justify integer data; justify decimal data on decimal point.
- ___ R 6 Long strings of numbers should be delimited with spaces, commas, or slashes; no leading zeros.
- ___ R 7 If a data record extends beyond a line, additional lines should be identified as continuations.
- ___ R 8 Tabular display data should be arranged to show similarities, differences, trends, or relationships.
- ___ R 9 Data that are important, require immediate response, and/or are frequent should be at the top.
- ___ R 10 If the data extend across several pages, related data should be placed on the same page.
- ___ R 11 The format for the data display should be compatible with the format used for data entry.
- ___ R 12 The ordering and layout of corresponding fields should be consistent and have the same names.

Section 8.5.1 Graphic Display Window Design

- ___ R 1 A graphic display window should be used for line graphs, bar charts, histograms, or flow charts.
- ___ R 2 The window should include a title and be sized so that the entire graphic display is visible.
- ___ R 3 Users should not have to scroll or resize a graphic display window to view its contents.

Section 8.5.1.1 Line Graphs

- ___ R 1 Line graphs should be used for trends and for spatial, time-critical, or imprecise information.
- ___ R 2 The axes of the graph should be clearly labeled, including the unit of measurement as appropriate.
- ___ R 3 The labels should be in upper- and lower-case letters and oriented left to right for normal reading.
- ___ R 4 Minimum and maximum values should be indicated on each axis, with up to 9 intermediate markings.
- ___ R 5 The starting point of an axis should be 0, with the gradations indicated in whole numbers.
- ___ R 6 Gradations should be standard intervals; intervening gradations consistent with the labeled scale.
- ___ R 7 Labels should be used instead of legends or keys to identify the data.
- ___ R 8 The labels should be oriented horizontally and located next to the data being referenced.
- ___ R 9 Each line or curve on a graph should be labeled and coded; critical or abnormal data should be coded.
- ___ R 10 Grid lines should be unobtrusive and not obscure the data presented in the graph.
- ___ R 11 Users should be able to display or suppress grid lines as desired.
- ___ R 12 The same coding scheme should be used consistently within an application.
- ___ R 13 Multiple trend lines should be presented on a single graph.
- ___ R 14 Users should be able to redraw multiple graphs using the same scale to facilitate comparison.
- ___ R 15 For precise values, users should be able to display the actual values on the graph and to zoom.

Section 8.5.1.2 Bar Charts and Histograms

- ___ R 1 Bar charts should be used to show several intervals, histograms to show a large number of intervals.
- ___ R 2 Bar graphs should have a consistent orientation; comparable bars should be adjacent.
- ___ R 3 When displayed data are compared with a critical value, a reference index should be provided.
- ___ R 4 The bar chart or histogram should be designed to conform to user expectations.
- ___ R 5 The use of icons to represent quantitative information should be avoided.
- ___ R 6 Charts and axes should be clearly labeled, and important information should be highlighted.
- ___ R 7 When bars are presented in pairs, they should be labeled as a unit and include a legend.

Section 8.5.1.3 Flow Charts

- ___ R 1 Flow charts should be used for a schematic representation of sequences or processes.
- ___ R 2 The steps in a flow chart should be presented in a logical order.
- ___ R 3 If there is no inherent logic, the steps should be ordered to minimize the size of the flow chart.
- ___ R 4 The path indicated in the flow chart should be left to right, top to bottom, or clockwise.
- ___ R 5 Each decision point in the flow chart should contain a single, simple decision.
- ___ R 6 The flow chart elements and lines should be consistently coded to assist in understanding.
- ___ R 7 The flow chart should provide directional indicators to indicate the sequence to be followed.
- ___ R 8 A legend should describe each element and code, and critical information and/or steps highlighted.

Version 1.1, Change 1

Section 8.5.2 Manipulation of Graphical Data

- ___ R 1 Users should be able to select/edit object attributes, change sizes, fill areas with colors/patterns.
- ___ R 2 Applications should automatically align, complete figures, draw lines between user-specified points.
- ___ R 3 Users should be able to draw objects such as lines, rectangles, ovals, and arcs.
- ___ R 4 Objects should emerge as they are drawn and be easy to reposition, duplicate, and delete.
- ___ R 5 Users should be able to group separate objects into a single object.
- ___ R 6 Objects should be displayed as they will be printed, or there should be an option to see this format.
- ___ R 7 Original graphics should be retained until users confirm that the objects are to be changed.
- ___ R 8 The objects should not be modified automatically as users change them.

Section 8.6.1 Map Window Design

- ___ R 1 A map window may contain the map and controls, or only the map with controls in dialog windows.
- ___ R 2 A map window should include identifying information about the map along with status information.
- ___ R 3 Identifying and status information may be in a window subarea or along the bottom window margin.
- ___ R 4 A continuous indicator of pointer location on the map should be available in a standard window area.
- ___ R 5 Maps should be displayed using the same orientation, and the important features should be labeled.
- ___ R 6 Labels should be positioned consistently with respect to features, not obscure, and remain legible.
- ___ R 7 When displaying color overlays, a color coding key should also be displayed.
- ___ R 8 Users should be able to display the coding key as desired without having to redisplay the overlay.
- ___ R 9 A dialog window displaying a coding key should be the minimum size and obscure little of overlay.
- ___ R 10 If appropriate, the key should be a scale so that users can interpret the coding in the overlay.
- ___ R 11 Standard symbology and color codes should be employed, with help available to users.
- ___ R 12 Symbols should be placed on the map accurately or connected to the map with graphics.
- ___ R 13 If multiple symbol sets are available, users should be able to select a set without losing data.
- ___ R 14 Symbols should be placed accurately or connected to location with arrows, lines, or graphics.
- ___ R 15 Symbol labels should appear next to the symbol and present essential information about it.
- ___ R 16 The background of the symbol and label should be transparent to not obscure other information.

Section 8.6.2 Map Manipulation

- ___ R 1 Users should be able to customize a map window to fit the task being performed.
- ___ R 2 Users should be able to pan and zoom a map as desired.
- ___ R 3 Position or change indicators should be provided to return quickly to the normal or starting map.
- ___ R 4 When a map is zoomed, the size of symbols, labels, and features should be adjusted to be readable.
- ___ R 5 Users should be able to define a baseline position on a map and return to this position quickly.
- ___ R 6 Users should be able to place window, define appearance, select objects, alter tactical appearance.
- ___ R 7 Users should be able to add, edit, reposition, delete, determine distance/bearing between points.
- ___ R 8 Latitude and longitude should be enterable to the level of accuracy needed.
- ___ R 9 Calculations (e.g., range, bearing, and position) should reflect accuracy appropriate to the scale.
- ___ R 10 Users should be able to select and deselect symbols on the map per table 2-2.
- ___ R 11 Users should be able to view or declutter, and obtain additional information for selected symbols.
- ___ R 12 The intensity of the map should be adjustable to fade out without losing all map features.
- ___ R 13 Users should be able to distinguish coincident point symbols and to obtain ambiguity information.

Section 8.6.3 Automatic Updating

- ___ R 1 Users should be able to select categories for automatic update and frequency and rate of update.
- ___ R 2 If appropriate, users should be able to temporarily stop and then resume further updates.

Section 8.7.1 Message Preparation

- ___ R 1 Message preparation windows should follow the same design as data entry windows.
- ___ R 2 Users should be given basic message header fields and supported in specifying the message address.

Version 1.1, Change 1

- ___ R 3 Option menus should be available for selecting from limited sets of frequently used terms.
- ___ R 4 When replying to a message, the appropriate addressee(s) should be provided automatically.
- ___ R 5 Users should be able to build/maintain lists of common addresses and to select from these lists.
- ___ R 6 Addresses should be checked prior to transmission; users should correct errors before sending.
- ___ R 7 Pre-formatted standard forms should be available; format control during entry should be automatic.
- ___ R 8 Users should be able to specify data, incorporate data files, save during preparation/completion.

Section 8.7.2 Message Transmission

- ___ R 1 Message transmission procedures should be designed to minimize the user actions required.
- ___ R 2 Users should be able to initiate message transmission directly (e.g., select a Transmit push button).
- ___ R 3 If a message cannot be sent immediately, it should be queued automatically.
- ___ R 4 Users should be able to assign message priorities and to cancel or abort a transmission.
- ___ R 5 Status feedback should be available, confirming messages sent and indicating transmission failures.
- ___ R 6 Users should be able to specify feedback wanted; an automatic log should be maintained.
- ___ R 7 Undelivered messages should be saved in the event of transmission failure.

Section 8.7.3 Message Receipt

- ___ R 1 Users should be informed when high priority messages are received.
- ___ R 2 Message notification should not interfere, but should provide some indication of urgency.
- ___ R 3 Incoming messages should be automatically queued by time and priority; logs should be maintained.
- ___ R 4 Users should be able to review summary information, display messages, save/file, and discard.
- ___ R 5 A message should be displayed in a text window; users should be able to scroll, save, and print it.
- ___ R 6 During login, a system may provide users with a list of new messages received.
- ___ R 7 During a session, a system may activate an alert indicator to inform users of priority messages.

INDEX

<u>Topic</u>	<u>Section</u>	<u>Page</u>
Abbreviations	7.3.3	7-14
Acronyms	7.3.3	7-14
Action icons	5.1	5-1
	6.2.1	6-2
	7.1.1.3.3	7-4
	7.2.2	7-13
Action menu options	4.1.2.1	4-1
	5.1.1	5-2
	Table 5-1	5-4
Alerts	6.2.1	6-2
	6.4.1	6-8
	7.4	7-17
	7.5.5	7-20
Apple interface guidelines	1.4	1-5
	1.7	1-7
Application-level help	8.1.1	8-1
Application windows	7.0	7-1
Organization	7.1	7-1
Design considerations	7.5	7-18
Army guidelines	1.7	1-8
	7.3.1	7-14
Arrow pointer	2.1.1.2	2-2
	Table 2-1	2-3
Assumptions	1.3	1-2
Audience	1.5	1-6
Auditory signals	7.5.5	7-20
	2.4.3	2-16
Automatic logout	6.5.2	6-10
Automatic updating	7.5.6	7-21
	8.6.3	8-18
Available menu options	4.1.2.4	4-4
Backward tabulation	2.1.2.3	2-6
Backward text deletion	2.1.2.3	2-7
	2.3	2-14
Bar charts	8.5.1.2	8-13
Capitalization	7.3.2	7-14

	7.5.5	7-20
Cascading menus	4.1.4	4-5
Case-sensitive search	5.5.3	5-14
Caution pointer	Table 2-1	2-3
Check buttons	5.3	5-9
	Table 7-1	7-10
	7.2.2	7-13
	7.5.1	7-18
	7.6.1	7-22
Checklist	Appendix B	B-1
Classification bar	6.2.1	6-2
	Table 6-1	6-3
	6.2.3	6-6
Classification markings	6.2.3	6-5
Closing windows	2.4.5	2-17
	3.3.6	3-6
Coding (see color coding)	7.5.5	7-20
Color coding	1.3.4	1-4
	7.4	7-17
	7.5.5	7-20
Color of objects	6.2.2	6-2
	Table 6-1	6-3
	7.2	7-9
	Table 7-1	7-10
	Appendix A	A-1
	Table A-1	A-3
	Table A-2	A-5
Color use	7.4	7-16
	8.4.2	8-9
Commercial Off-the-Shelf (COTS) software	1.3.3	1-3
	1.6	1-6
	6.4.1	6-8
	8.3.2	8-8
	8.5.2	8-13
	Appendix A	A-1
Compliance	1-6	1-6
Control panels	7.1.1.3.1	7-2
	7.5.1	7-18
Controls	5.0	5-1
	7.1.1.3.1	7-2
	7.5.3	7-19
	8.2.1	8-6
Critical information	7.5.5	7-20

Data entry and manipulation (see text entry)	8.2.3	8-6
	8.7.1	8-18
Data entry windows (see message preparation windows)	8.2	8-4
Data field format (see text entry field)	8.2.2	8-6
Data query	8.2.4	8-7
Date-time clock	6.2.1	6-2
	2.1.2.3	2-7
	2.2.3	2-9
Default push buttons	5.1.2	5-2
	7.1.1.3.2	7-3
Default settings	6.4.2	6-9
	7.6.1	7-22
DoD Human-Computer Interface Style Guide	1.4	1-4
	1.8	1-8
	6.2.3 footnote	6-5
Dialog windows (see modal and modeless dialog windows)	3.4.1	3-6
	3.4.2.1	3-8
	Table 5-1	5-7
	7.1.1	7-1
	7.5.1	7-18
	7.6.4	7-23
Direct manipulation	2.1	2-1
Disabling user input	2.4.2	2-15
Drag-and-drop	2.2.4	2-9
	2.2.4.1	2-11
Dynamic information in windows	7.5.6	7-21
	8.6.3	8-18
Editable text	5.4.2	5-11
	Table 7-1	7-10
	7.3.5	7-15
Editing commands	2.2.4.2	2-12
	Table 5-1	5-5
	5.4.3	5-11
Environmental considerations	1.2	1-1
Ergonomic specifications	1.2	1-1
Error detection	2.4.4	2-16
Error feedback	2.4.4	2-16
	8.2.3	8-6
Error message windows	7.1.2.3	7-7
Exclusive settings	5.2	5-8

Explicit destruction	2.4.5	2-16
Explicit window focus	3.2.1	3-3
Extended selection	2.2.3	2-9
F1 function key	2.1.2.5	2-8
File manipulation	Table 5-1	5-4
	6.4.1	6-9
Flow charts	8.5.1.3	8-13
Font	7.3.1	7-13
Forward tabulation	2.1.2.3	2-6
Forward text deletion	2.1.2.3	2-7
	2.3	2-14
Function key assignments	2.1.2.5	2-8
Gadgets	Appendix A	A-1
Government Off-the-Shelf (GOTS) software	1.3.3	1-3
Graphic display windows	8.5	8-11
Hardware configuration	1.3.1	1-2
Help	2.1.2.5	2-8
	Table 5-1	5-7
	6.3.2	6-6
	6.3.4	6-7
	7.1.1.3.2	7-3
	8.1.1	8-1
	8.2.1	8-6
Help windows	8.1	8-1
Histograms	8.5.1.2	8-13
Hooking algorithms	2.1.3	2-8
Hourglass pointer	Table 2-1	2-3
I-beam pointer	Table 2-1	2-3
	2.1.2.2	2-5
Icon box	3.3.4 footnote	3-5
Icons	3.2.3	3-3
	3.3.4	3-5
	Table 7-1	7-10
Iconified windows	3.2.3	3-3
	3.3.4	3-5
	7.5.6	7-21

	7.6.5	7-23
Iconifying windows	3.3.4	3-5
IEEE user interface drivability recommendations	1.4	1-5
	1.8	1-8
	2.1.2.3 footnote	2-6
	2.1.2.5 footnote	2-8
	2.2.2 footnote	2-9
	2.2.4.1 footnote	2-12
	2.4.3	2-16
	7.5.5	7-21
Illumination	1.2 footnote	1-1
Immediate feedback	2.4.2	2-14
	7.1.1.4	7-5
	8.2.3	8-7
Implicit input focus	3.2.1 footnote	3-3
Incremental search in list boxes	5.5.3	5-14
Information message windows	7.1.2.4	7-8
Information-only area	5.4.2	5-11
	8.2.1	8-4
Input devices	2.1	2-1
Input devices, alternative	2.1.3	2-8
Input focus	3.2.1	3-3
Insert mode	2.3	2-14
Interactive control	2.4	2-14
Keyboard accelerators	2.2.4	2-11
	4.1.3	4-4
	Table 4-1	4-6
	7.2.2	7-12
Keyboard assignments	2.1.2.3	2-6
	2.1.2.4	2-7
	2.1.2.5	2-8
Keyboard functions	2.1.2.1	2-4
Keyboard input	2.1	2-1
Keyboard traversal	7.2.1	7-12
	Appendix A	A-1
Lighting	1.2	1-1
Lightpens	2.1.3	2-8
Line graphs	8.5.1.1	8-11
List boxes (see multi-column list boxes)	5.5	5-12
	Table 7-1	7-10

	7.2.2	7-13
	7.5.1	7-18
Login	6.1	6-1
Login window	6.1.2	6-1
Logout	6.5	6-9
Lowering windows	3.3.5	3-5
Mandatory provisions	1.7	1-7
Map windows	8.6	8-16
Maximize button	3.1	3-1
	3.3.3	3-4
	Table 7-1	7-10
Memory load	7.5.4	7-19
Menu options	4.1	4-1
Menus (see pop-up menus, pull-down menus, option menus)	4.0	4-1
Message handling windows	8.7	8-18
Message preparation	8.7.1	8-18
Message receipt	8.7.3	8-20
Message transmission	8.7.2	8-19
Message windows	3.4.2.2	3-8
	7.1.2	7-5
	7.4	7-16
Military Handbook 761A	1.4	1-5
	1.8	1-8
	2.4.3	2-15
	7.5.5	7-21
Military Standard 12	7.3.3	7-14
Military Standard 411	7.3.3	7-14
Military Standard 783	7.3.3	7-14
Military Standard 1472D	1.4	1-5
	1.8	1-8
	4.1.2.4 footnote	4-4
Military Standard 1801	1.4	1-5
	1.8	1-8
	2.1.1.1	2-2
Minimize button	3.1	3-1
	3.3.4	3-5
	Table 7-1	7-10
Modal/modeless dialog windows	3.4.2.1	3-8
	7.1.1.3.2	7-3
	7.6.4	7-23
Modifier key assignments	2.1.2.4	2-7

Motif gadgets	Appendix A	A-1
Motif graphical user interface	1.3.1	1-2
	1.5	1-6
	1.6	1-7
	7.2.1	7-12
	7.3.1	7-14
Motif resource settings	Appendix A	A-1
	1.7	1-7
	6.2.2	6-2
	7.2.1	7-9
Motif Style Guide	Appendix A	A-1
	1.3.1 footnote	1-2
	1.4	1-5
	1.7 footnote	1-7
	1.8	1-8
	2.1 footnote	2-1
	2.1.1.2 footnote	2-2
	2.1.2.3 footnote	2-6
	2.2.2 footnote	2-9
	2.2.4.1	2-12
	3.1 footnote	3-2
	3.3.4 footnote	3-5
	3.4.1 footnote	3-6
	4.1.2.4 footnote	4-4
	Table 4-1 footnote	4-6
	4.1.5 footnote	4-8
	Table 5-1 footnote	5-4
	5.5.3 footnote	5-14
Motif widgets	Table 7-1 footnote	7-11
	Appendix A	A-1
	Table A-2	A-5
Motif window manager	Appendix A	A-1
	Table A-1	A-3
	Table A-2	A-5
Mouse	1.3.1	1-2
	2.1	2-1
Moving pointer	Table 2-1	2-3
Moving windows	3.3.2	3-4
Multi-column list boxes	5.5.5	5-16
	8.4.1	8-8
NIST reference model	1.4	1-4
NATO symbology	8.6.1	8-17

Navigation aids	6.3.5	6-7
	8.1.1	8-1
Navigation key assignments	2.1.2.3	2-6
Noise	1.2	1-1
Noneditable text	5.4.2	5-11
	Table 7-1	7-10
	7.3.4	7-15
Nonexclusive settings	5.3	5-9
NTDS symbology	8.6.1	8-17
Object-action selection paradigm	2.2.1	2-8
Object appearance and behavior	6.2.2	6-2
	Table 6-1	6-3
	7.2	7-9
	Table 7-1	7-10
Object color	Table 6-1	6-3
	7.2	7-9
	Table 7-1	7-10
	Appendix A	A-1
	Table A-1	A-3
	Table A-2	A-5
Object manipulation	2.2.4	2-9
Object selection and deselection	2.2.2	2-9
	Table 2-2	2-10
Object size	7.2.2	7-12
Online training	8.1.4	8-4
OPNAVINST 5510.1H	6.2.3 footnote	6-5
Option menus	4.3	4-8
	Table 7-1	7-11
	7.5.1	7-18
	8.2.1	8-4
	8.7.1	8-18
Overlays	8.6.2	8-17
Placement of windows	3.3.1	3-4
	7.6.2	7-22
Pointer	2.1.1.1	2-1
	2.1.1.2	2-2
	Table 2-1	2-3
Pointing device	2.1.1	2-1
Pop-up menus	4.2	4-8
	Table 7-1	7-11

	7.5.1	7-18
POSIX	1.3.1	1-2
Primary windows (see action vocabulary, window components, window organization)	3.4.1 7.5.1	3-6 7-18
Printing	6.4.1 8.3.2 8.5.2	6-8 8-8 8-15
Progress message windows	7.1.2.1	7-6
Pull-down menus (see keyboard accelerators)	4.1 7.1.1.2 Table 7-1 7.2.2 8.2.1	4-1 7-2 7-11 7-12 8-6
Purpose	1.2	1-1
Push buttons (see action icons)	5.1 Table 5-1 7.1.1.3.2 Table 7-1 7.1.2 7.2.2 7.5.1 8.1.1 8.2.1	5-1 5-4 7-3 7-10 7-6 7-13 7-18 8-1 8-6
Query language	8.2.4	8-7
Question message windows	7.1.2.5	7-9
Radio buttons	5.2 Table 7-1 7.2.2 7.5.1 7.6.1	5-8 7-10 7-19 7-18 7-22
Readability of text	7.3.1 7.3.4	7-13 7-15
Recommended provisions	1:7	1-7
Redundant coding	1.3.4 7.4	1-4 7-17
References	1.8	1-7
Resize border	3.1 Table 7-1	3-1 7-10
Resize pointer	Table 2-1	2-3

Resizing windows	3.3.3	3-4
	7.5.3	7-19
	7.6.3	7-23
Response time	2.4.3	2-15
Routing menu options	4.1.2.1	4-1
Scales	5.7	5-19
	Table 7-1	7-11
Scroll bars	5.6	5-16
	Table 7-1	7-11
In list boxes	5.5	5-12
In resizable windows	7.6.3	7-23
In text entry fields	5.4	5-10
Search capability in text windows	8.3.2	8-8
Search/replace capability in text windows	8.3.2	8-8
Secondary windows	3.1	3-1
	3.4.1	3-6
	3.4.2	3-8
Selection	2.2	2-8
In list boxes	5.5.4	5-14
In pull-down menus	4.1.5	4-7
Objects	2.2.2	2-9
	Table 2-2	2-10
Selection, other types	2.2.3	2-9
Selection dialog box	5.5.3	5-14
Setting menu items	4.1.2.1	4-1
Sighting pointer	Table 2-1	2-3
Size of font	7.3.1	7-13
Size of objects	7.2.2	7-12
Size of text	7.3.1	7-13
Size of windows	7.6.2	7-22
Software configuration	1.3.1	1-2
Space insertion	2.1.2.3	2-7
Special Appeal Board	1.6	1-7
Speed search in list boxes	5.5.3	5-14
Support functions	6.4.1	6-8
Symbology	8.6.1	8-17
System functionality	1.3.2	1-2
System integration	1.3.1	1-3
System-level help	8.1.1	8-1
System login	6.1	6-1
System logout	6.5	6-9
System menu	6.3	6-6

	8.1.1	8-1
System start-up	6.1.3	6-2
System support	6.4	6-8
System users	1.3.4	1-4
	6.4.2	6-9
System window	6.2	6-2
	Table 6-1	6-3
Tabular data windows	8.4	8-8
Tabulation	2.1.2.3	2-6
Text (see color of objects, editable text, noneditable text)	7.3	7-13
Text deletion	2.1.2.3	2-7
Text documents	8.3.1	8-7
	8.3.2	8-8
Text entry (see data entry and manipulation)	2.1.2.2	2-5
	2.3	2-14
	5.4.3	5-11
	8.2.3	8-6
Text entry fields (see data entry fields, noneditable text)	2.1.2.2	2-5
	2.1.2.3	2-6
	5.4	5-9
	Table 7-1	7-10
	7.2.2	7-13
	8.2.1	8-4
Text insertion cursor (see text entry)	2.1.2.2	2-4
	2.1.2.3	2-6
	5.4.4	5-12
Text manipulation	8.3.2	8-8
Text windows	8.3	8-7
Trackball	1.3.1	1-2
	2.1	2-1
Toggle menu options	4.1.2.1	4-2
Touchscreens	2.1.3	2-8
Unavailable menu options	4.1.2.4	4-4
Undo capability	2.4.6	2-17
Unix	1.3.1	1-2
User-computer interaction	2.0	2-1
User control of interaction	2.4.1	2-14
User-initiated logout	6.5.1	6-9
User-specified settings		

Users	1.3.4	1-4
	6.4.2	6-9
Utilities	6.3.3	6-7
	6.4.1	6-8
Variable action buttons	2.1.3	2-8
Visual feedback for errors	2.4.2	2-15
Vocabulary	5.1.1	5-2
	Table 5-1	5-4
Voice input	2.1.3	2-8
Warning message windows	7.1.2.2	7-7
Widgets	Appendix A	A-1
	Table A-2	A-5
Window components	3.1	3-1
	Table 7-1	7-10
Window design considerations	7.5	7-18
Window hierarchies	3.4.1	3-6
Window management	7.6	7-22
Window manipulation (see placement of windows, resizing windows)	3.3	3-4
Window Menu	3.1	3-1
	Table 7-1	7-10
Window Menu button	3.1	3-1
	Table 7-1	7-10
Window organization (see window design considerations)	7.1	7-1
Window placement	3.3.1	3-4
	7.6.2	7-22
Window size	3.3.1	3-4
	7.6.2	7-22
Window states	3.2	3-3
Window types (see dialog windows, message windows, primary windows, secondary windows)	3.4	3-6
Workstation design	1.2	1-1
	1.8	1-9
X pointer	2.1.1.2	2-2
X Windows	1.3.1	1-2
	2.1.1.2	2-2