

HFES 200

HUMAN FACTORS ENGINEERING OF SOFTWARE USER INTERFACES



Published by
Human Factors and Ergonomics
Society P.O. Box 1369
Santa Monica, CA 904-6-1369 USA
310/394-1811, Fax 310/394-2410
info@hfes.org, <http://hfes.org>

© Copyright 2000–2008, Human Factors and Ergonomics Society. All rights reserved.
No portion of this document may be reproduced, stored, or transmitted by any means
without prior written permission of the Human Factors and Ergonomics Society.

Master Table of Contents

Preface 1

Part 1: Introduction to the HFES 200 Project..... 6

Part 2: Accessibility 16

Part 3: Interaction Techniques..... 114

Part 4: Interactive Voice Response (IVR) and Telephony..... 342

Part 5: Visual Presentation and Use of Color 366

Copyright 2008 by Human Factors and Ergonomics Society

Preface

The Human Factors and Ergonomics Society Human Factors Engineering of Software User Interfaces (now known as “HFES 200”) represents the culmination of many years of effort from an extraordinarily talented and tenacious group of leading human factors professionals.

HFES 200 is a major landmark in the field of human-computer interaction (HCI).

Recognition is due first and foremost to the editors of the four major technical sections constituting the standard: Professor Greg Vanderheiden, Dr. James Williams, Professor Daryle Gardner-Bonneau, and Dr. Aaron

The four technical sections of the standard were authored by exceptional editors who defined design guidelines capturing human-computer-interaction-based research findings, established best practices, and consensus of international experts. This standard is a remarkable achievement particularly because of its meticulous harmonization with international efforts in the International Organization for Standardization (ISO), and the accompanying wide-ranging issues that had to be addressed and resolved.

The editors...

Gregg Vanderheiden

Professor, Industrial & Systems Engineering and Biomedical Engineering Director Trace R&C Center, University of Wisconsin - Madison
Cochair, Coeditor W3C Web Content Accessibility Guidelines
Chair, V2 Technical Group of INCITS (International Committee on Information Technology Standards / ITI)
Member Working Group 5 ISO TC 159
Member ISO/IEC JTC1 SWG-A Special working group on accessibility
Member SC35
Member COST119ter

Aaron Bangor is a Principal Member of Technical Staff at AT&T Labs, Inc., in Austin, Texas. There he researches the usability and accessibility of user interfaces for communication services. Dr. Bangor also serves on the Texas Governor's Committee on People with Disabilities. He is a member of the U.S. Technical Advisory Group to ISO TC 159 / SC1 and SC 4 and joined the HFES 200 HCI Standards committee in 2004. He assumed editorship of Part 5 in 2005 and has also contributed to Parts 2 and 4.

Daryle Gardner-Bonneau - Principal, Bonneau and Associates." I am also an Adjunct Associate Professor of Speech Pathology and Audiology at Western Michigan University. My ANSI-related standards activities are: Member, Human Engineering Committee, Association for the Advancement of Medical Instrumentation (AAMI) - was involved in the development of ANSI/AAMI HE-74 and am currently involved in the development of ANSI/AAMI HE-75 (human engineering process standard and guidelines for the design of medical devices, respectively). With respect to ISO, my activities consist of the following: Chair, U.S. TAG to TC159 - Ergonomics; U.S. Expert to ISO TC159 WG2 - Ergonomic Requirements for People with Special Needs; U.S. Expert to ISO TC159 SC4 WG5 - Software Ergonomics and Human-Computer Dialogues; U.S. Expert to ISO TC159 SC5 WG5 - Environments for People with Special Requirements.

© Copyright 2000–2008, Human Factors and Ergonomics Society. All rights reserved. No portion of this document may be reproduced, stored, or transmitted by any means without prior written permission of the Human Factors and Ergonomics Society.

HFES 200

James Williams is Principal Ergonomist for Synergetic Applications, a consultancy specializing in software ergonomics. Dr. Williams has been in technical and management positions in various areas of human performance for over 40 years. He has been involved with ANSI and ISO standards for the last 20 years and chairs the US Technical Advisory Groups to ISO TC159/SC1 and SC4. In addition, he is the Deputy Convener of TC159/SC4/WG1 and a member of SC4 Working Groups 5 and 6. Dr. Williams has been involved with the HFES 200 Committee since its beginning and is editor of Part 3 and a contributor to Part 2.

HFES 200 is nothing less than a monumental effort from an extraordinarily dedicated and talented group of human factors professionals who contributed tens of thousands of hours of volunteer efforts over a period of more than 15 years. HFES 200 was initially established in 1986 to explore the prospects for developing a set of software user interface design requirements and recommendations rigorously based on empirical evidence and established best practices, and which could be consistently verified through usability testing methods. John Karat chaired the committee from 1986 until 1994 and provided outstanding leadership to facilitate the committee's formative contributions to ISO standards activities in Software Ergonomics (ISO TC 159 "Ergonomics"/SC4 "Displays and Controls"/WG5 "Human System Interaction").

The members of this committee created the formative user interface design recommendations and requirements that comprised the foundation of the International Organization for Standardization (ISO) standards for Dialog Techniques, Visual Presentation, Direct Manipulation, and others. Perhaps more important, the active participation and contributions from this committee to ISO software ergonomics standards efforts firmly established the critical importance of rigorous methodologies and objective, empirical research to substantiate human-computer interaction standards.

There are many members of the HFES 200 Committee who have delivered sustained contributions to this important national and international effort for many years. I would like to use this opportunity to recognize several key individuals whose contributions have exerted a formative influence on the development of software ergonomics standards.

Jim Williams has consistently delivered major contributions to HFES 200 and other standards activities for over 15 years, and he continues to serve as editor of Part 3: Interaction Techniques. This document addresses a major area of human-computer interaction by providing hundreds of design recommendations for a wide range of interaction methods including: visual menus, command languages, direct manipulation, user guidance, dialog boxes and forms, and others. HFES 200 developed documents in each of these topic areas, and each of these individual documents reflects a major effort in itself with scores of references and many committee reviews. Jim Williams has contributed enormously to the HFES 200 effort, as well as other HFES, ANSI, and ISO standards activities, by leading the development of the Interaction Techniques document originally developed in this committee and eventually established as the very first ISO standard in the area of Software Ergonomics and User Interfaces. Jim put in nothing less than a heroic effort in conducting exhaustive literature research, documenting hundreds of supporting empirical studies and citations, and enduring endlessly grueling reviews both nationally and internationally for many, many years. He deserves an ultra-marathon medal!

Eric Bergman pioneered a major new area of standards activity by writing the first HFES standards document specifically focused on Accessibility of Software User Interfaces. Eric

© Copyright 2000–2008, Human Factors and Ergonomics Society. All rights reserved. No portion of this document may be reproduced, stored, or transmitted by any means without prior written permission of the Human Factors and Ergonomics Society.

HFES 200

compiled a wide range of results from research and other material to create a comprehensive set of design recommendations and requirements for Software Accessibility that ultimately became an ISO standard, and which continues to serve as the foundations for the current HFES 200 version. Shawn Henry contributed important extensions and refinements to Software Accessibility before it was distributed for review by an ANSI Canvass Committee in 2000. Paul Reed served as editor for the comments received during the Canvass review. Due to procedural issues, Software Accessibility was unfortunately not approved as an ANSI standard in 2001.

Daryle Gardner-Bonneau has also expanded HFES standards in a new direction by completing the specifications for HFES 200 Part 4: Voice Input/Output and Telephony. She worked with a sub-group of telecommunications professionals to derive design recommendations reflecting research results and best practices for IVR, and it is expected that this effort will be expanded in the future to address Voice Input/Output and Auditory Codes in the future.

Arnie Lund addressed an important domain of user interface design by providing comprehensive design recommendations for Visual Presentation and Use of Color. An extensive and wide-ranging effort to review relevant literature on the use of color in user interface design resulted in highly innovative design guidance. Beth Marshburn served as editor as this material was reviewed by the committee. Harry Blanchard is currently serving as the editor for this material.

Evelyn Williams provided major contributions with the initial versions of User Guidance (later completed by Jim Williams) and the complete Dialog Boxes and Forms sections of Part 3, and assiduously reviewed large quantities of research that reflected the latest developments in the area.

Pamela Jamar, Pat Billingsley, and Karen Rafnel provided a key contribution in the area of Menu Behavior and Appearance, and their work became an important element in ISO standards as well as being incorporated in the current version of HFES 200.

Elizabeth Buie, with a keen instinct for linguistic rigor and meticulous precision, tackled the difficult challenge of creating a Glossary and Definitions section that spanned a very broad range of technical areas and perspectives. She aptly handled the difficult position of being in “no man’s land” while various section editors pleaded their case for having their definition version used. Her efforts resulted in an exhaustive and consistent set of terms across the broad expanse of material in HFES 200.

Andrew Cohill and John Karat developed a Design Process specification that addressed a user-centered approach to interface design in the system development life cycle that served as a key influential contribution to ISO Software Ergonomics activities. John Karat also provided critical leadership for the HFES 200 Committee from 1987 through 1994 and ensured that efforts were appropriately focused contributions to key ISO standards activities.

I would also like to acknowledge the extended and ongoing support of HFES Executive Director Lynn Strother, the Technical Standards Committee Chairs Claire Gordon, Bob Beaton, and Ken Holdaway, and the members of the HFES Executive Council over the years.

© Copyright 2000–2008, Human Factors and Ergonomics Society. All rights reserved. No portion of this document may be reproduced, stored, or transmitted by any means without prior written permission of the Human Factors and Ergonomics Society.

HFES 200

Daryle Gardner-Bonneau has also expanded HFES standards in a new direction by completing the specifications for HFES 200 Part 4: Voice Input/Output and Telephony. She worked with a sub-group of telecommunications professionals to derive design recommendations reflecting research results and best practices for IVR, and it is expected that this effort will be expanded in the future to address Voice Input/Output and Auditory Codes in the future.

Arnie Lund addressed an important domain of user interface design by providing comprehensive design recommendations for Visual Presentation and Use of Color. An extensive and wide-ranging effort to review relevant literature on the use of color in user interface design resulted in highly innovative design guidance.

Harry Blanchard is currently serving as the editor for this material and we plan to include this material in the next version of HFES 200 under the ANSI Canvass Method.

Beth Marshburn served as editor as this material was reviewed by the committee.

Evelyn Williams provided major contributions with the initial versions of User Guidance (later completed by Jim Williams) and the complete Dialog Boxes and Forms sections of Part 3, and assiduously reviewed large quantities of research that reflected the latest developments in the area.

Pamela Jamar, Pat Billingsley, and Karen Rafnel provided a key contribution in the area of Menu Behavior and Appearance, and their work became an important element in ISO standards as well as being incorporated in the current version of HFES 200.

Elizabeth Buie, with a keen instinct for linguistic rigor and meticulous precision, tackled the difficult challenge of creating a Glossary and Definitions section that spanned a very broad range of technical areas and perspectives. She aptly handled the difficult position of being in “no man’s land” while various section editors pleaded their case for having their definition version used. Her efforts resulted in an exhaustive and consistent set of terms across the broad expanse of material in HFES 200.

Andrew Cohill and John Karat developed a Design Process specification that addressed a user-centered approach to interface design in the system development life cycle that served as a key influential contribution to ISO Software Ergonomics activities. John Karat also provided critical leadership for the HFES 200 Committee from 1987 through 1994 and ensured that efforts were appropriately focused contributions to key ISO standards activities.

I would also like to acknowledge the extended and ongoing support of HFES Executive Director Lynn Strother; the Technical Standards Committee Chairs Claire Gordon, Bob Beaton, and Ken Holdaway; and the members of the HFES Executive Council over the years.

Contributors to journal papers published about the HFES 200 Committee software ergonomics standards efforts include Elizabeth Buie, Daryle Gardner-Bonneau, James Carter, Jeff Fox, Ken Holdaway, Arnie Lund, Eric Bergman, Ken Holdaway, Scott Isensee, and Jim Williams.

© Copyright 2000–2008, Human Factors and Ergonomics Society. All rights reserved. No portion of this document may be reproduced, stored, or transmitted by any means without prior written permission of the Human Factors and Ergonomics Society.

Current active members of the committee include: Eric Bergman, Harry Blanchard, Elizabeth Buie, Daryle Gardner-Bonneau, James Carter (with his uniquely Canadian perspective), Shawn Henry, Scott Isensee, Arnie Lund, Paul Reed, Evelyn Williams, and Jim Williams. For the past several years, Harry Blanchard has served as Co-Chair, Daryle Gardner-Bonneau has served as Secretary, and Evelyn Williams has served as Librarian.

Previous members of the committee who provided important contributions to our standards document and activities include: Gary Beck, Pat Billingsley, Andrew Cohill, Thom Foote-Lennox, Jeff Fox, Shawn Henry, Ken Holdaway, Pamela Jamar, John Karat, Beth Marshburn, Jackie Schrier, and Karen Rafnel.

The challenging and diligent work completed by this committee has formed the basis for the content of the ISO Software Ergonomics standards today and, perhaps more importantly, established the importance of having a rigorous methodology for defining, prioritizing, and implementing software ergonomics standards.

It has been a tremendous privilege and honor, not to mention just a lot of plain fun, to work with such a talented, distinguished, and good-humored group of professionals, and it has been an outstanding opportunity for us all to contribute to the improvement of the quality of user experiences both nationally and internationally.

With kindest regards,
Paul Reed
HFES 200 Chair

© Copyright 2000–2008, Human Factors and Ergonomics Society. All rights reserved. No portion of this document may be reproduced, stored, or transmitted by any means without prior written permission of the Human Factors and Ergonomics Society.

HFES 200

HFES 200

HUMAN FACTORS ENGINEERING OF SOFTWARE USER INTERFACES

PART 1: INTRODUCTION

© Copyright 2000–2008, Human Factors and Ergonomics Society. All rights reserved. No portion of this document may be reproduced, stored, or transmitted by any means without prior written permission of the Human Factors and Ergonomics Society.

HFES 200.1

Copyright Human Factors and Ergonomics Society
Provided by IHS under license with HFES
No reproduction or networking permitted without license from IHS
Full document details: [HFES 200 2008](#)

Licensee=, User=,
Not for Resale,

 **Documents**
Document Sharing Hub

Part 1 Table of Contents

1	Introduction to the HFES 200 Project	8
2	Scope of HFES 200.....	9
3	Organization of HFES 200	9
3.1	Recommendations Clause.....	10
3.2	Informative Annex A: Sample Procedure for Assessing Compliance and Adherence.....	10
3.3	HFES 200.2: Accessibility	10
3.4	HFES 200.3: Interaction Techniques	11
3.5	HFES 200.4: Interactive Voice Response.....	12
3.6	HFES 200.5: Visual Presentation and Use of Color	12
4	Application of HFES 200 in Analysis, Design and Evaluation.....	12
4.1	Introduction to the Use of HFES 200 Design Recommendations	12
4.2	Audience of HFES 200	13
4.3	Analysis	13
4.4	Design.....	14
4.5	Evaluation.....	14
4.6	Claims of Conformance	14

© Copyright 2000–2008, Human Factors and Ergonomics Society. All rights reserved. No portion of this document may be reproduced, stored, or transmitted by any means without prior written permission of the Human Factors and Ergonomics Society.

HFES 200.1

1 Introduction to the HFES 200 Project

The Human Factors and Ergonomics Society undertook the HFES 200 project following extensive participation in the development of international standards through the International Organization for Standardization (ISO). HFES 200 committee members were primary contributors of material to the ISO standards development efforts. Central principles driving the HFES 200 Committee's contributions to ISO were the use of user interface design guidance based on robust empirical findings and established, proven industry practices. The objective of the HFES 200 standard is to consolidate available design guidance to provide design recommendations that will lead to usability benefits such as increased ease of learning and ease of use of software, and accessibility benefits such as increased compatibility of assistive technology with available Operating System software.

The HFES 200 Software User Interface standard consists of five parts:

- HFES 200.1: Introduction
- HFES 200.2: Accessibility
- HFES 200.3: Interaction Techniques
- HFES 200.4: Interactive Voice Response (IVR)
- HFES 200.5: Visual Presentation and Use of Color

A primary objective of HFES 200 Part 3 and Part 5 is to be compatible with ISO 9241, Parts 12 through 17, which provide software user interface design recommendations for dialog techniques, such as menus and direct manipulation, and for presentation of information. Only a small fraction of the ISO recommendations have changed significantly in content. Appendix B in Part 3 and Part 5 identifies the disposition of every corresponding ISO recommendation as each is treated in HFES 200.

HFES 200 Part 2: Accessibility and Part 4: Interactive Voice Response consist of new material that has not been published previously, whereas Part 3: Interaction Techniques and Part 5: Visual Presentation and Use of Color incorporate material from approved International Organization for Standardization (ISO) 9241 (1996–1998) Part 12 through Part 17.

The goal of this introduction is to:

- a) Outline the content of the HFES 200 parts,
- b) Explain the relationships among the individual parts of HFES 200, and
- c) Provide guidance on the relevance of individual parts to the development process — that is, to help readers understand when and how to use the parts of HFES 200.

© Copyright 2000–2008, Human Factors and Ergonomics Society. All rights reserved. No portion of this document may be reproduced, stored, or transmitted by any means without prior written permission of the Human Factors and Ergonomics Society.

HFES 200.1

The ultimate beneficiaries of HFES 200 are the end users of software. It was the needs of these users that motivated the recommendations in HFES 200. The application of this standard is intended to provide user interfaces that are more usable, accessible, and consistent and that enable greater productivity and satisfaction.

2 Scope of HFES 200

The scope of HFES 200 is primarily focused on user interaction with software for personal, business, and educational use, most commonly implemented on a desktop PC or terminal (e.g., see ANSI/HFS 100-1988). Most of the recommendations in this standard also apply to home and mobile computing and to interactive voice response applications. Many of the recommendations may also apply to other interactive software.

Although the recommendations in this standard do not definitively address areas such as high-risk applications, nuclear power plant control room environments, alarm/security applications, process control, and entertainment, many of the recommendations in HFES 200 can be used to improve the quality of software applications in these environments. Developers in these areas are advised to obtain guidance from more directly applicable standards or guideline documents.

This standard does not address system or software design processes. For guidance on the role of software ergonomics in the system/software design process, please refer to ISO 9241 Part 11, Guidance on Usability, and to ISO 13407, Human-Centred Design for Interactive Systems (1997).

3 Organization of HFES 200

The specification or design of human-computer interaction (also known as human-computer dialogues) should be driven by an understanding of the requirements of the users, their tasks, and the business functions to be performed. Many trade-offs typically need to be evaluated, including technology drivers and constraints. There are usually several technology and design options, and the final choice may be influenced by the desire for consistency (e.g., with existing practices or across a range of systems). Decisions about the overall suitability of a software user interface can be assisted by reference to standards containing general principles of dialogue design, such as ISO 9241 Part 10. Specific recommendations for dialogue design techniques are addressed in HFES 200 Part 3, Interaction Techniques.

The recommendations in this standard explain how each recommendation should be applied. Individual recommendations should be evaluated for their applicability and, if

© Copyright 2000–2008, Human Factors and Ergonomics Society. All rights reserved. No portion of this document may be reproduced, stored, or transmitted by any means without prior written permission of the Human Factors and Ergonomics Society.

HFES 200.1

judged to be applicable, should be implemented in the relevant dialogue technique unless there is evidence that to do so would cause deviation from the design objectives or would result in an overall degradation in usability.

3.1 Recommendations Clause

The recommendations clause within each Part of HFES 200 provides specific design guidance for readers. The clauses in Parts 3 and 4 based on ISO 9241 contain the recommendations of the particular Part of ISO 9241. Many recommendations within HFES 200 are conditional recommendations (or "if" statements). Often the "if" statement refers to the context of use (e.g., in HFES 200 Part 3, "If appropriate for the task, in deeply structured menus (more than three levels), users should be provided with the capability to go from one Part (node) of the structure to another without returning to the initial common node."

3.2 Informative Annex A: Sample Procedure for Assessing Compliance and Adherence

Annex A within selected parts of HFES 200 provides an example of a procedure for determining whether the applicable recommendations in the particular Part of HFES 200 have been met. It should be noted that the procedure described is provided as guidance and is not a rigid process to be used as a substitute for the standard itself. This procedure provides a two-stage process for determining (a) which recommendations are relevant and (b) whether those relevant recommendations have been followed.

3.3 HFES 200.2: Accessibility

The recommendations in HFES 200 Part 2 focus on features and functions of computer operating systems, drivers, application services, other software layers upon which applications depend, and software applications, that increase the accessibility of applications for users with disabilities. Hardware is not specifically addressed by any recommendations; however, various hardware assistive devices may exploit recommended functions that are provided by operations system and application software. Accessibility of applications is often facilitated by a combination of both hardware and software. The guidelines on accessibility are aimed at reducing the need for add-on assistive technologies (hardware and software) while promoting increased usability of systems using such add-on technologies when they are required. The proposed standard on accessibility does not address the behavior or requirements for assistive technologies themselves (including assistive software) but, instead, focuses on

© Copyright 2000–2008, Human Factors and Ergonomics Society. All rights reserved. No portion of this document may be reproduced, stored, or transmitted by any means without prior written permission of the Human Factors and Ergonomics Society.

enabling features provided by operating systems, drivers, application services, and other software upon which applications depend.

In part 2 (only) of this standard the provisions are assigned to three levels of priority. This was done in order to provide those wishing to prioritize accessibility interventions with information as to which provisions had the greatest importance (e.g. if not done it would be difficult or impossible to make the software accessible even with assistive technologies). It also provided harmonization with ISO 9241-171 which divides its provisions into SHALL and SHOULD. Level 1 in HFES 200.2 corresponds to 9241-171 SHALLs. [The ISO standards corresponding to HFES 200 Parts 3, 4, and 5 do not have SHALL provisions] Three (rather than two) levels were provided in HFES 200.2 due to the number of provisions that would fall into level 2 and the fact that three levels provided a more informative profile of priority. It should be noted that provisions at all three levels are important to different individuals and that these provisions do not constitute all that can be done to make software more accessible to all types, degrees and combinations of disability.

While extensive efforts have been made to harmonize HFES 200 Part 2 with the corresponding ISO 9241-171 standards project, one important distinction is that HFES 200 technical guidance is exclusively in the form of non-mandatory recommendations, while the ISO document includes many mandatory “shall” requirements. Another important difference is that HFES 200 Part 2 contains three “levels” of technical guidance in order to communicate the prioritization of recommendations for developers. In contrast, ISO 9241-171 contains only two levels: 1) Requirements (“shalls”) and 2) Recommendations (“shoulds”). Because the ISO 9241-171 document is still undergoing ISO due process and is subject to change, it is not feasible to make definitive statements about the relationship of HFES 200 to ISO 9241-171.

3.4 HFES 200.3: Interaction Techniques

The recommendations in this Part of HFES 200 include material from ISO 9241 parts 14 through 17. Please note that HFES 200 parts 1 through 5 contain no requirements. All ISO 9241 requirements included in HFES 200 have been changed to recommendations.

The ISO 9241 parts included in HFES 200 Part 3 are

- Menu Dialogues (ISO 9241-14),
- Command Dialogues (ISO 9241-15),
- Direct Manipulation Dialogues (ISO 9241-16), and
- Form Filling Dialogues (ISO 9241-17).

The recommendations in HFES 200 Part 3 address each of the interaction techniques (also known as dialogue techniques) listed above. In menu dialogues, the dialogue

© Copyright 2000–2008, Human Factors and Ergonomics Society. All rights reserved. No portion of this document may be reproduced, stored, or transmitted by any means without prior written permission of the Human Factors and Ergonomics Society.

system presents one or more groups of options to the user, the user chooses one or more options, and the computer executes the process denoted by the selected option or options. In command dialogues, users input (by recall) either complete or abbreviated command phrases as required by the command language syntax, and the computer performs the actions associated with the commands and their parameters. Direct manipulation dialogues enable users to perform operations by acting on displayed objects in ways analogous to manipulating physical entities. In form filling dialogues, users fill in, select entries for, or modify labeled fields on an area of the display.

3.5 HFES 200.4: Interactive Voice Response

The recommendations in HFES 200 Part 4 consists of completely new material developed by the HFES 200 committee, and provide ergonomic design recommendations for the design of voice-based user interfaces for telephony-based Interactive Voice Response (IVR) systems. Guidance is provided on input techniques, error handling, human operator access, prompting, messages, output techniques, and other topics.

3.6 HFES 200.5: Visual Presentation and Use of Color

HFES -200 Part 5 incorporates material from ISO 9241 Part 12: Presentation of Information, and includes substantial new material developed by the HFES 200 Committee addressing the Use of Color in user interface design. Recommendations are provided on topics such as organization of Information, graphical objects, coding of information (e.g. graphical coding and alphanumeric coding), and color selection.

4 Application of HFES 200 in Analysis, Design and Evaluation

4.1 Introduction to the Use of HFES 200 Design Recommendations

Interaction design depends upon the task, the user, the environment and the available technology. Consequently, the five parts of HFES 200 cannot be applied without knowledge of the design and use context of the interface. The parts of HFES 200 are not intended to be used as a prescriptive set of rules to be applied in their entirety. Rather, it is assumed that the designer has proper information available concerning task and user requirements and understands the use of available technology. (Note that this process could require consultation with a qualified interface design and usability professional as well as empirical testing with real users.) HFES 200 can be applied at

© Copyright 2000–2008, Human Factors and Ergonomics Society. All rights reserved. No portion of this document may be reproduced, stored, or transmitted by any means without prior written permission of the Human Factors and Ergonomics Society.

HFES 200.1

various stages of the development process. The intended users of HFES 200 and the use of the particular parts of HFES 200 during analysis, design, and evaluation are described in the following paragraphs.

4.2 Audience of HFES 200

The following are examples of the intended audience of this standard:

- (a) designers of user interfaces, who will apply the standard during the development process,
- (b) designers and developers of user interface development tools to be used by designers of user interfaces,
- (c) designers and developers of operating systems, drivers, application services, and other software layers upon which applications depend.
(NOTE: When designing an operating system, driver, application service, or user interface development tool, one should follow the applicable recommendations of HFES 200 in order to ensure that applications and user interface tools utilizing these elements are capable of producing user interfaces that meet the recommendations of HFES 200.)
- (d) designers of user interface style guides to be used by designers of user interfaces,
(NOTE: HFES 200 is not a user interface style guide. User interface style guides typically apply to a given operating system or specific software development projects. HFES 200 does not consider a specific operating system or a specific application domain. However, the applicable recommendations of HFES 200 should be applied when designing a user interface style guide for a particular operating system or a particular software development project. Any recommendation given in a user interface style guide should conform to the corresponding recommendation in HFES 200.)
- (e) the buyer, who will reference this Part of the standard during the product procurement process,
- (f) evaluators responsible for ensuring products meet the recommendations in HFES 200, and
- (g) end users who will gain from the potential benefits provided by the standard.

Designers using HFES 200 need to know that the interface they are developing meets the recommendations provided in the individual parts of HFES 200. Likewise, buyers and evaluators need a means to determine whether or not a product matches the recommendation. It is not intended that every recommendation should be applied, only those that are relevant.

4.3 Analysis

An important prerequisite to analysis is determining the intended users and the "context of use" of the interface. Knowledge concerning the various interaction or dialogue

© Copyright 2000–2008, Human Factors and Ergonomics Society. All rights reserved. No portion of this document may be reproduced, stored, or transmitted by any means without prior written permission of the Human Factors and Ergonomics Society.

techniques and user guidance will also be useful in identifying user performance considerations at this stage.

4.4 Design

HFES 200 should be used as a source of information for making design trade-offs and solving design problems. In each part, the recommendations concerning the detailed design of dialogues should be applied during the design of that particular dialogue or interaction. Individual recommendations should be evaluated for their applicability and, if judged to be applicable, should be implemented unless there is evidence that to do so would cause deviation from the design objectives or would result in an overall degradation in usability. In judging whether or not applicable recommendations have been met, evaluators should evaluate the product or observe representative users of the product in the context of accomplishing the user's tasks via the dialogue or interaction system. Sample procedures that support the determination of applicability and judgments of whether a recommendation has been followed are provided in Informative Annex A of selected HFES 200 parts.

4.5 Evaluation

The evaluation procedure should be based on an analysis of typical users, their typical and critical tasks, and their typical use environment. HFES 200 can be used for evaluation by checking to see if each applicable guideline in appropriate parts is met. Applicability is determined by considering any conditional "if" statement in the guidelines and appropriate constraints in the design environment. Applicability can be determined by using system documentation analysis, documented evidence, observations, analytical evaluation, and/or empirical evaluation. When a guideline is applicable, it is necessary to determine whether or not it has been met. Adherence can be determined by using measurements, observations, documented evidence, and analytical and/or empirical evaluation. A sample procedure for evaluating a user interface with respect to the recommendations given in HFES 200 parts is described in detail in Informative Annex A in certain parts of HFES 200.

4.6 Claims of Conformance

If a product is claimed to have met the applicable recommendations in any Part of HFES 200, the procedures used in establishing requirements for, developing, and/or evaluating the recommendations shall be specified. The level of specification of the procedure is a matter of negotiation between the involved parties.

Reporting guidelines are provided by ANSI-NCITS 354-2001: Industry Usability Reporting and the Common Industry Format.

© Copyright 2000–2008, Human Factors and Ergonomics Society. All rights reserved. No portion of this document may be reproduced, stored, or transmitted by any means without prior written permission of the Human Factors and Ergonomics Society.

HFES 200.1

References

- Human Factors and Ergonomics Society. (1988). *American national standard for human factors engineering of visual display terminal workstations* (ANSI/HFS 100-1988). Santa Monica, CA: Author.
- International Organization for Standardization. (1996). *Ergonomic requirements for office work with visual display terminals: Part 10. Dialogue principles* (ISO 9241-10). Geneva: Author.
- International Organization for Standardization. (1997a). *Ergonomic requirements for office work with visual display terminals: Part 14. Menu dialogues* (ISO 9241-14). Geneva: Author.
- International Organization for Standardization. (1997b). *Ergonomic requirements for office work with visual display terminals: Part 15. Command dialogues* (ISO 9241-15). Geneva: Author.
- International Organization for Standardization. (1998a). *Ergonomic requirements for office work with visual display terminals: Part 11. Guidance on usability* (ISO 9241-11). Geneva: Author.
- International Organization for Standardization. (1998b). *Ergonomic requirements for office work with visual display terminals: Part 12. Presentation of information* (ISO 9241-12). Geneva: Author.
- International Organization for Standardization. (1998c). *Ergonomic requirements for office work with visual display terminals: Part 13. User guidance* (ISO 9241-13). Geneva: Author.
- International Organization for Standardization. (1998d). *Ergonomic requirements for office work with visual display terminals: Part 17. Form-filling dialogues* (ISO 9241-17). Geneva: Author.
- International Organization for Standardization. (1999a). *Ergonomic requirements for office work with visual display terminals: Part 16. Direct manipulation dialogues* (ISO 9241-16). Geneva: Author.
- International Organization for Standardization. (1999b). *Human-centred design processes for interactive systems* (ISO 13407). Geneva: ISO Technical Committee ISC/TC 159, Ergonomics Subcommittee SC4, Ergonomics of Human System Interaction.

HFES 200

HUMAN FACTORS ENGINEERING OF SOFTWARE USER INTERFACES

PART 2: ACCESSIBILITY

© Copyright 2000–2008, Human Factors and Ergonomics Society. All rights reserved. No portion of this document may be reproduced, stored, or transmitted by any means without prior written permission of the Human Factors and Ergonomics Society.

HFES 200.2

Copyright Human Factors and Ergonomics Society
Provided by IHS under license with HFES
No reproduction or networking permitted without license from IHS
Full document details: [HFES 200 2008](#)

Licensee=/, User=,
Not for Resale,

Part 2 Table of Contents

1 Introduction	22
2 Scope	24
3 Normative References	25
4 Terms and Definitions.....	25
5 Rationale and Benefits of Implementing Accessibility	33
6 Sources of Variation in User Characteristics	33
7 How to Use This Standard.....	34
7.1 General	34
7.2 Conformance	35
8 General Guidelines.....	36
8.1 Names and labels for user interface elements	36
8.1.1 Provide a name for each user interface element (Level 1).....	36
8.1.2 Provide meaningful names (Level 2)	37
8.1.3 Provide unique names within context (Level 3)	38
8.1.4 Make names available to assistive technology (Level 1).....	39
8.1.5 Display names (Level 2)	39
8.1.6 Provide names and labels that are short (Level 2).....	40
8.1.7 Provide text label display option for icons (Level 3)	40
8.1.8 Properly position the labels of user interface elements on screen (Level 3).....	40
8.2 User preference settings	40
8.2.1 Enable individualization of user preference settings (Level 2)	40
8.2.2 Enable adjustment of attributes of common user interface elements (Level 2).....	41
8.2.3 Enable individualization of the user interface look and feel (Level 3).....	42
8.2.4 Enable individualization of the cursor and pointer (Level 1).....	42
8.2.5 Provide user-preference profiles (Level 3)	42
8.2.6 Provide capability to use preference settings across locations (Level 3).....	43
8.2.7 Enable user control of timed responses (Level 1)	44
8.3 Special considerations for accessibility adjustments.....	44
8.3.1 Make controls for accessibility features discoverable and operable (Level 1)	44
8.3.2 Safeguard against inadvertent activation or deactivation of accessibility features (Level 2)	45
8.3.3 Avoid interference with accessibility features (Level 1).....	45
8.3.4 Inform user of accessibility feature on/off status (Level 2)	45
8.3.5 Inform user of accessibility feature activation (Level 2).....	45
8.3.6 Enable persistent display (Level 2).....	45
8.4 General control and operation guidelines	46
8.4.1 Enable switching of input/output alternatives (Level 3)	46
8.4.2 Optimize the number of steps required for any task (Level 3).....	46
8.4.3 Provide “Undo” and/or “Confirm” functionality (Level 2)	47
8.4.4 Provide alternatives when assistive technology is not operable (Level 1)	48
8.4.5 Enable software-controlled media extraction (Level 1).....	48
8.4.6 Support “Copy” and “Paste” operations (Level 2).....	48
8.4.7 Support “Copy” operations in non-editable text (Level 3).....	49
8.4.8 Enable selection of elements as an alternative to typing (Level 2)	49
8.4.9 Allow warning or error information to persist (Level 1)	50
8.4.10 Present user notification using consistent techniques (Level 2).....	50
8.4.11 Provide understandable user notifications (Level 2).....	50
8.4.12 Facilitate navigation to the location of errors (Level 2)	51
8.5 Compatibility with assistive technology	51

© Copyright 2000–2008, Human Factors and Ergonomics Society. All rights reserved. No portion of this document may be reproduced, stored, or transmitted by any means without prior written permission of the Human Factors and Ergonomics Society.

HFES 200.2

Copyright Human Factors and Ergonomics Society
 Provided by IHS under license with HFES
 No reproduction or networking permitted without license from IHS
 Full document details: HFES 200 2008

Licensee=, User=,
 Not for Resale,

8.5.1 General	51
8.5.2 Enable communication between software and assistive technology (Level 1).....	51
8.5.3 Use standard accessibility services (Level 1).....	52
8.5.4 Make user interface element information available to assistive technologies (Level 1).....	53
8.5.5 Allow assistive technology to change keyboard focus and selection (Level 1)	55
8.5.6 Provide user interface element descriptions (Level 1).....	56
8.5.7 Make event notification available to assistive technologies (Level 1)	56
8.5.8 Allow assistive technology to access resources (Level 2).....	57
8.5.9 Use system-standard input/output (Level 1).....	58
8.5.10 Enable appropriate presentation of tables (Level 1).....	59
8.5.11 Accept the installation of keyboard and/or pointing device emulators (Level 1)	59
8.5.12 Allow assistive technology to monitor output operations (Level 1).....	59
8.5.13 Support combinations of assistive technologies (Level 3).....	60
8.6 Closed systems	60
8.6.1 Read content on closed systems (Level 1).....	60
8.6.2 Announce changes on closed systems (Level 1)	60
8.6.3 Operable through tactilely discernable controls (Level 1).....	60
8.6.4 Pass through of system functions (Level 1).....	61
9 Inputs	61
9.1 Alternative input options	61
9.1.1 Provide keyboard input from all standard input mechanisms (Level 2).....	61
9.1.2 Provide parallel keyboard control of pointer functions (MouseKeys) (Level 1)	61
9.1.3 Provide pointer control of keyboard functions (Level 2)	62
9.1.4 Provide speech recognition services (Level 2)	62
9.1.5 Provide system-wide spell checking tools (Level 3)	62
9.2 Keyboard focus	63
9.2.1 Provide keyboard focus and text cursor (Level 1)	63
9.2.2 Provide highly visible keyboard focus and text cursors (Level 1).....	63
9.2.3 Restore state when regaining keyboard focus (Level 2)	64
9.3 Keyboard input	64
9.3.1 General	64
9.3.2 Enable full use via keyboard (Level 1).....	64
9.3.3 Enable sequential entry of multiple (chorded) keystrokes (StickyKeys) (Level 1).....	66
9.3.4 Provide adjustment of delay before key acceptance (SlowKeys) (Level 1).....	67
9.3.5 Provide adjustment of same-key double-strike acceptance (BounceKeys) (Level 1)	67
9.3.6 Provide adjustment of key repeat rate (Level 2).....	68
9.3.7 Provide adjustment of key-repeat onset (Level 2).....	68
9.3.8 Allow users to turn key repeat off (Level 1)	68
9.3.9 Provide notification about toggle-key status (ToggleKeys) (Level 2)	68
9.3.10 Provide accelerator keys (Level 3)	69
9.3.11 Provide implicit or explicit designators (Level 2).....	70
9.3.12 Reserve accessibility accelerator key assignments (Level 1)	71
9.3.13 Enable remapping of keyboard functions (Level 3)	71
9.3.14 Separate keyboard navigation and activation (Level 1).....	72
9.3.15 Follow platform keyboard conventions (Level 2)	72
9.3.16 Facilitate list and menu navigation (Level 2)	73
9.3.17 Facilitate navigation of controls by grouping (Level 3)	73
9.3.18 Arrange controls in task-appropriate navigation order (Level 3)	73
9.3.19 Allow users to customize accelerator keys (Level 3).....	74
9.4 Pointing devices	74
9.4.1 General	74
9.4.2 Provide direct control of pointer position from external devices. (Level 1).....	74
9.4.3 Provide easily-selectable pointing device targets (Level 2).....	74

© Copyright 2000–2008, Human Factors and Ergonomics Society. All rights reserved. No portion of this document may be reproduced, stored, or transmitted by any means without prior written permission of the Human Factors and Ergonomics Society.

9.4.4	Enable reassignment of pointing device button functions (Level 1)	75
9.4.5	Provide alternative input methods for complex pointing device operations (Level 2)	75
9.4.6	Enable pointing device button-hold functionality (Level 1)	75
9.4.7	Provide adjustment of delay of pointing-device button-press acceptance (Level 2)	76
9.4.8	Provide adjustment of minimum drag distance (Level 2)	77
9.4.9	Provide adjustment of multiple-click parameters (Level 1)	77
9.4.10	Provide adjustment of pointer speed (Level 1)	77
9.4.11	Provide adjustment of pointer acceleration (Level 1)	78
9.4.12	Provide adjustment of pointer movement direction (Level 3)	78
9.4.13	Provide a means for finding the pointer (Level 1)	78
9.4.14	Provide alternatives to simultaneous pointer operations (Level 1)	78
10	Outputs	79
10.1	General output guidelines	79
10.1.1	Avoid seizure-inducing flash rates (Level 1)	79
10.1.2	Enable user control of time-sensitive presentation of information (Level 1)	79
10.1.3	Provide accessible alternatives to task relevant audio and video (Level 1)	80
10.2	Visual output (displays)	80
10.2.1	Enable users to adjust graphic attributes (Level 3)	80
10.2.2	Provide a visual information mode usable by users with low visual acuity (Level 3)	81
10.2.3	Use text characters as text, not as drawing elements (Level 3)	81
10.2.4	Provide keyboard access to information displayed outside the physical screen (Level 1)	81
10.3	Text/Fonts	82
10.3.1	Do not convey information by visual font attribute alone (Level 2)	82
10.3.2	Enable user to set minimum font size (Level 3)	82
10.3.3	Adjust the scale and layout of user interface elements as font-size changes (Level 2)	83
10.4	Color	83
10.4.1	Do not convey information by color output alone (Level 1)	83
10.4.2	Provide color schemes designed for people with disabilities (Level 3)	83
10.4.3	Provide Individualization of color schemes (Level 2)	84
10.4.4	Allow users to individualize color coding (Level 2)	84
10.4.5	Provide contrast between foreground and background (Level 2)	85
10.5	Window appearance and behavior	85
10.5.1	Provide unique and meaningful window titles (Level 2)	85
10.5.2	Provide window titles that are unique within the windowing system (Level 2)	85
10.5.3	Enable non-pointer navigation to windows (Level 1)	86
10.5.4	Enable “always on top” windows (Level 1)	86
10.5.5	Provide user control of multiple “always on top” windows (Level 1)	87
10.5.6	Enable user choice of effect of pointer and keyboard focus on window stacking order (Level 3)	87
10.5.7	Enable window positioning (Level 1)	87
10.5.8	Enable window resizing (Level 2)	87
10.5.9	Support minimize, maximize, restore and close windows (Level 2)	88
10.5.10	Enable windows to avoid taking focus (Level 1)	88
10.6	Audio output	89
10.6.1	Use tone pattern rather than single tone to convey information (Level 2)	89
10.6.2	Enable control of audio volume (Level 1)	89
10.6.3	Use a mid-frequency range for non-speech audio (Level 3)	89
10.6.4	Enable adjustment of audio output (Level 3)	89
10.6.5	Control of background and other sound tracks (Level 3)	90
10.6.6	Use specified frequency components for audio warnings and alerts (Level 3)	90
10.6.7	Allow users to choose visual alternative for audio output (ShowSounds) (Level 1)	90
10.6.8	Synchronize Audio equivalents for visual events (Level 1)	91
10.6.9	Provide speech output services (Level 1)	91

© Copyright 2000–2008, Human Factors and Ergonomics Society. All rights reserved. No portion of this document may be reproduced, stored, or transmitted by any means without prior written permission of the Human Factors and Ergonomics Society.

10.7 Text equivalents of audio (captions)	91
10.7.1 Display any captions provided (Level 1)	91
10.7.2 Enable system-wide control of captioning (Level 2)	92
10.7.3 Support system settings for captioning (Level 1)	92
10.7.4 Position captions to not obscure content (Level 2)	92
10.8 Media	92
10.8.1 Enable users to stop, start and pause media (Level 1)	92
10.8.2 Enable users to replay, rewind, pause and fast or jump forward (Level 2)	92
10.8.3 Allow user to control presentation of multiple media streams (Level 2)	93
10.8.4 Update equivalent alternatives for media when the media changes (Level 1)	93
10.9 Tactile Output	93
10.9.1 Do not convey information by tactile output alone (Level 2)	93
10.9.2 Use familiar tactile patterns (Level 2)	94
10.9.3 Enable tactile output to be adjusted (Level 2)	94
11 Online Documentation, Help, and Support Services	94
11.1 Documentation and Help	94
11.1.1 Provide understandable documentation and Help (Level 2)	94
11.1.2 Provide user documentation and Help in accessible electronic form (Level 1)	94
11.1.3 Provide text alternatives in electronic documentation and Help (Level 1)	95
11.1.4 Write instructions and Help without unnecessary device references (Level 2)	95
11.1.5 Provide documentation and Help on accessibility features (Level 1)	95
11.2 Support services	96
11.2.1 Provide accessible support services (Level 1)	96
11.2.2 Provide accessible training materials (Level 2)	96
Appendix A (Informative): Issues Regarding Activity Limitations	97
A.1 General	97
A.2 Sensory Functions	97
A.2.1 Vision	97
A.2.1.1 Individuals who are unable to see	97
A.2.1.2 Individuals with low vision	98
A.2.2 Hearing	99
A.2.2.1 Individuals who are unable to hear	99
A.2.2.2 Individuals with a reduced ability to hear	100
A.2.3 Tactile	100
A.3 Neuromusculoskeletal and movement related functions	100
A.3.1 General	101
A.3.2 Individuals with limitations in motor activity	101
A.3.3 Physical Size and Reach	101
A.3.4 Speech Disabilities	101
A.4 Mental functions	102
A.4.1 General	102
A.4.2 Limitations on attention	102
A.4.3 Limitations on memory	102
A.4.4 Limitations on the mental functions of language	103
A.5 Individuals with other disabilities	103
A.5.1 Allergy	103
A.5.2 Other Functional Limitations	103
A.6 Multiple body function effects	103
Appendix B (Informative): StickyKeys, SlowKeys, BounceKeys, FilterKeys, MouseKeys, RepeatKeys, ToggleKeys, SoundSentry, ShowSounds and SerialKeys	104

© Copyright 2000–2008, Human Factors and Ergonomics Society. All rights reserved. No portion of this document may be reproduced, stored, or transmitted by any means without prior written permission of the Human Factors and Ergonomics Society.

Introduction.....	104
Permission to Use Terms.....	104
Description of Access Features.....	105
Common activation behaviors	105
StickyKeys	105
SlowKeys	106
BounceKeys.....	107
FilterKeys	107
MouseKeys	108
RepeatKeys	109
ToggleKeys.....	109
SoundSentry	110
ShowSounds.....	110
Time Out (For All Access Features)	111
SerialKeys.....	111
Appendix C (Informative): Bibliography.....	111

© Copyright 2000–2008, Human Factors and Ergonomics Society. All rights reserved. No portion of this document may be reproduced, stored, or transmitted by any means without prior written permission of the Human Factors and Ergonomics Society.

HFES 200.2

Copyright Human Factors and Ergonomics Society
 Provided by IHS under license with HFES
 No reproduction or networking permitted without license from IHS
 Full document details: HFES 200 2008

Licensee=/, User=,
 Not for Resale,

1 Introduction

The purpose of this American National Standard is to specify design guidance for human-system software interfaces to provide the highest level of accessibility for as many users as possible. A primary goal of these design guidelines is human-system interfaces with increased usability that promote increased effectiveness, efficiency, and satisfaction for people who have a wide variety of capabilities and preferences. Accessibility is closely related to the concept of usability, and the properties of these concepts overlap and intertwine in so many ways among various applications, tasks, devices, and users, that the boundary between them is unavoidably fuzzy.

These design guidelines focus on user-system interfaces and interfaces between operating systems, middleware, application software, and other software layers, to facilitate the development of interfaces to systems and products that are intended for use by people with the widest range of capabilities. It is desirable to define accessibility goals and features for a particular product as early as possible in the design process so that developments costs can be reduced compared to modifying products for accessibility after they have been design.

The design guidelines in this document are primarily based on the current understanding of users with the widest range of capabilities and limitations. These users include individuals who: 1) have particular sensory, motor and/or cognitive impairments, 2) have limitations as a result of aging or disease processes, and 3) are affected by environments that may limit normal sensory, motor, or cognitive functioning.

Therefore, accessibility addresses a widely defined group of users including:

- people with physical, sensory and cognitive impairments present at birth or acquired during life,
- elderly people who can benefit from new products and services but experience reduced physical, sensory and cognitive capacities,
- people with temporary disabilities, such as a person with a broken arm or someone who has forgotten his/her glasses, and
- people who experience difficulties in particular situations, such as a person who works in a noisy environment or has both hands occupied by other work.

When designing and evaluating human system interfaces other terms are often used that are associated with accessibility.

The North American term “Universal Design” is similar to the European term ‘Design for All’ that identify the goal of enabling maximum access to the maximum number and diversity of users, irrespective of their skill level, language, culture, environment or

© Copyright 2000–2008, Human Factors and Ergonomics Society. All rights reserved. No portion of this document may be reproduced, stored, or transmitted by any means without prior written permission of the Human Factors and Ergonomics Society.

disability. However, it is always true that there will be a minority of disabled people with severe or multiple impairments who will need adaptations or specialized products. The term accessibility as defined in this standard emphasizes the twin goals of: 1) maximizing the number of users and 2) striving to increase the level of usability that these users experience.

The design requirements and recommendations in this document are specified for user interface system design, appearance and behavior. The primary objective is to allow software to be used by as broad an audience as possible. It is recognized that some users of software will need assistive devices in order to use a system. Therefore, this Standard includes, in the concept of accessibility, the capability of a system to provide connections to and enable successful interaction with assistive technologies. Guidance is provided on designing software that integrates as effectively as possible with common assistive technologies (e.g. speech synthesizers, Braille input and output devices), when they are utilized.

In addition, this standard addresses the increasing need to consider social and legislative demands to ensure accessibility by removing barriers that prevent people with special requirements from participating in life activities including the use of environments, services, products and information. Designing software-user interfaces for accessibility increases the number of people who can use systems by taking into account the varying physical and sensory capabilities of user populations. Designing for accessibility benefits disabled users by providing features that enable them to use software that would otherwise be inaccessible, and also by making software easier for them to use.

Many accessibility features also benefit users who do not have a disability, by enhancing usability and providing additional individualization possibilities. They may also help to overcome a temporary defect (e.g. a broken arm or hand). They benefit designers and suppliers by expanding the number of potential users (and thus sales for their products) and often by making the product compliant with legal requirements for accessibility. They benefit companies buying software by expanding the number of employees who may use the software.

It is important to note that accessibility may be provided by a combination of both hardware and software. Assistive technologies typically provide specialized input and output capabilities not provided by the system. Software examples include on-screen keyboards that replace physical keyboards, screen-magnification software that allows users to view their screen at various levels of magnification, and screen-reading software that allows blind users to navigate through applications, determine the state of controls, and read text via text-to-speech conversion. Hardware examples include head-mounted pointers that replace mice and Braille output devices that replace a video display. There are many other examples not listed here. When users provide add-on assistive software and/or hardware, usability is enhanced to the extent that systems and

© Copyright 2000–2008, Human Factors and Ergonomics Society. All rights reserved. No portion of this document may be reproduced, stored, or transmitted by any means without prior written permission of the Human Factors and Ergonomics Society.

HFES 200.2

Copyright Human Factors and Ergonomics Society
Provided by IHS under license with HFES
No reproduction or networking permitted without license from IHS
Full document details: HFES 200 2008

Licensee=, User=,
Not for Resale,

applications integrate with those technologies. For this reason, operating systems need to provide “hooks”, Application Programming Interfaces (APIs), programmatic access, or other features to allow software to operate effectively with add-on assistive software and hardware as recommended in this standard. If systems do not provide support for assistive technologies, the probability increases that users will encounter problems with compatibility, performance and usability.

The ultimate beneficiary of this Standard will be the end user of the software. Although it is unlikely that the end-users will read this document, its application by designers, developers, buyers and evaluators should provide user interfaces that are more accessible. This standard concerns the development of software for user interfaces. However, those involved in designing the hardware aspects of user interfaces may also find the standard useful.

Note: In this document the term ‘developers’ is used as a shorthand for all those involved in the development of software design and creation which sometimes can span different collaborating or contracting organizations.

2 Scope

The scope of HFES 200.2 is primarily focused on the design of accessible software for personal, educational, business, and public use, most commonly implemented on a desktop computer or a client/server combination (e.g., see ANSI/HFS 100 DSTU, 2003). Most of the recommendations in this standard also apply to home and mobile computing and to interactive voice response applications. Many of the recommendations may also apply to other software. Although the recommendations in this standard do not definitively address areas such as high-risk applications, nuclear power plant control room environments, alarm/security applications, process control, and entertainment, many of the recommendations in HFES 200 can be used to improve the quality of applications in these environments. Developers in these areas are advised to obtain guidance from more directly applicable standards or guideline documents.

The standard promotes increased usability of systems for a wider range of users. While it does not cover the behavior or requirements for assistive technologies (including assistive software), it addresses the use of assistive technologies as an integrated component of interactive systems.

This part of the standard is intended for use by those responsible for the specification, design, development, evaluation and procurement of software operating systems and software applications. It is meant to be used in concert with the other parts of this standard.

3 Normative References

The following standards contain provisions that, through reference in this text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below.

ISO 9241-10: Ergonomic requirements for office work with visual display terminals (VDTs) - Part 10: Dialogue principles

ISO 13407 Human-centered design processes for interactive systems

HFES 200.3: Human Factors Engineering of Software User Interfaces – Interaction Techniques

HFES 200.4: Human Factors Engineering of Software User Interfaces – Interactive Voice Response (IVR) and Telephony

HFES 200.5: Human Factors Engineering of Software User Interfaces – Visual Presentation and Use of Color

4 Terms and Definitions

For the purposes of this document, the following terms and definitions apply.

4.1 Accelerator Keys: Keys, key sequences, or key combinations which invoke an action immediately without displaying intermediate information (such as menus) or requiring pointer movement or any other user activity

NOTE: Also called shortcut keys and hot keys.

EXAMPLE: Users can often invoke the OK button by pressing ENTER, and the Cancel button by pressing ESC.

4.2 Accessibility Feature: feature (etc) that is specifically designed to increase the usability of products for those experiencing disabilities

4.3 Activation: Initiation of an action associated with a selected user interface element

© Copyright 2000–2008, Human Factors and Ergonomics Society. All rights reserved. No portion of this document may be reproduced, stored, or transmitted by any means without prior written permission of the Human Factors and Ergonomics Society.

HFES 200.2

4.4 Assistive Technology: Hardware or software that is added to or incorporated within a system that increases accessibility for an individual

EXAMPLES: Braille displays, screen readers, screen magnification software and eye tracking devices are assistive technologies.

4.5 Chorded Key-press: Keyboard key or pointing device button presses where more than one button is held down simultaneously to invoke an action

NOTE: This includes both uses of modifier keys with other (non-modifier) keys as well as use of multiple non-modifier keys to enter data or invoke an action.

4.6 Closed System: system that does not allow user connection or installation of assistive technology that would have programmatic access to the full user interface

NOTE This can be caused by policy, system architecture, physical constraints or any other reason.

4.7 Color Scheme: set of color assignments used for rendering user interface elements

NOTE: Color refers to a combination of hue, saturation, and brightness.

4.8 Contrast: In a perceptual sense, assessment of the difference in appearance of two or more parts of a field seen simultaneously or successively

Examples Brightness contrast, lightness contrast, color contrast, etc.

NOTE: Adapted from CIE 17.4 definition 845-02-47

4.9 Cursor: Visual indication of where the user interaction via keyboard (or keyboard emulator) will occur.

NOTE 1 Keyboard focus cursors and text cursors are types of cursors.

NOTE 2 Contrast with keyboard focus cursor (4.20), text cursor (4.32), and contrast with pointer (4.28).

4.10 Developers: Individuals or organizations that design and/or create software.

4.11 Explicit designator: A code or abbreviation for a menu option or control label, set apart (usually to the left) from the name, and typed in for selection.

NOTE: Contrast with “implicit designator.”

© Copyright 2000–2008, Human Factors and Ergonomics Society. All rights reserved. No portion of this document may be reproduced, stored, or transmitted by any means without prior written permission of the Human Factors and Ergonomics Society.

Example: In the following menu the explicit designators are “O”, “C”, “S”, and “P.”:

- O Open
- C Close
- S Save
- P Print

4.12 **Focus Cursor:** Indicator showing which user interface element has keyboard focus

NOTE 1: Also called “location cursor”. See also **Input focus** (4.16) and **Cursor** (4.9).

NOTE 2: The appearance of this indicator usually depends on the kind of user interface element that has focus. The user interface element with focus can be activated if it is a control (e.g. button, menu item) or selected if it is a selectable user interface element (e.g. icon, list item).

EXAMPLE: A box or highlighted area around a text field, button, list or menu options can serve as a focus cursor.

4.13 **Icon:** A graphic on a visual display terminal that represents an object or action of the user's task.

4.14 **Implicit Designator:** Portion of an option name or control label used for a keyboard selection

NOTE: Also called access key

EXAMPLE 1: In the portion of a menu shown in figure 1, the implicit designators are the underlined letters T,S,E,W, g, m, L, and D in top-to-bottom order. (Note that in “Large Icons”, the “g” is underlined.)

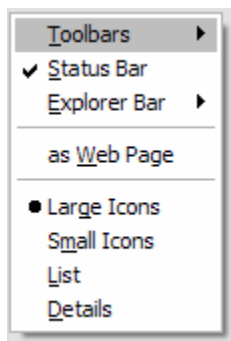


Figure 1

© Copyright 2000–2008, Human Factors and Ergonomics Society. All rights reserved. No portion of this document may be reproduced, stored, or transmitted by any means without prior written permission of the Human Factors and Ergonomics Society.

EXAMPLE 2: In the pushbuttons in figure 2, the implicit designators are D and P

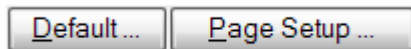


Figure 2

4.15 Individualization: Modification of interaction and presentation of information to suit individual capabilities and needs of users

4.16 Input Focus: Current assignments of the input from an input device to a user interface element

EXAMPLES: Pointer focus and keyboard focus are input foci.

4.17 Keyboard Emulator: Software or hardware that generates input that is identical to that which comes from a keyboard.

NOTE: A keyboard emulator may provide a representation of keys (e.g. on-screen keyboard) or it may not (e.g. voice recognition).

EXAMPLE: Platform-based on-screen keyboards, speech input, and handwriting are all examples of keyboard emulators if their output appears to applications as keystroke input.

4.18 Keyboard Equivalent: Key or key combination that provides access to a function usually activated by a pointing device, voice input, or other input or control mechanism.

4.19 Keyboard Focus: Current assignment of the input from the keyboard or equivalent to a user interface element

NOTE: For an individual user interface element focus is indicated by a focus cursor.

4.20 Keyboard Focus Cursor: visual indication of where the user interaction via keyboard (or keyboard emulator) will occur

Note: Contrast with keyboard focus (4.19), pointer (4.28), and text cursor (4.32).

4.21 Label: short, descriptive title for a user interface element (object).

NOTE 1 Labels include, but are not limited to, headings, prompts for entry fields, text or graphics that accompany and identify controls (e.g. displayed on the face of buttons) and audible prompts used by interactive voice response systems.

NOTE 2: Contrast with 4.25 “Name”, In this section label refers to the presented title for a user interface element. It contrasts with the Name attribute which may or may not be presented to users but is available to assistive technologies. Textual labels are often a visual display of the name.

EXAMPLE 1: On a screen used for initiating a print job, the control label is displayed as “Print”, and the underlined “P” indicates that “P” is the implicit designator for the “Print” command.

EXAMPLE 2:

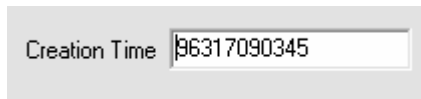


Figure 3 Text field with label.

EXAMPLE 3:

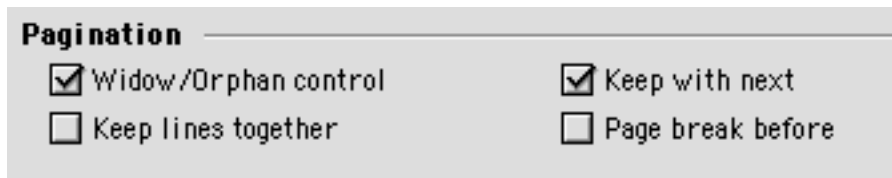


Figure 4. Check box group, with a label for the group and a label for each check box.

EXAMPLE 4: A window displays the image of a printer that the user can click to print the current document. This image’s label is image, its name attribute is “Print”, its role attribute is “Push Button”, and its description attribute is “A printer”.

See also **Name** (4.25)

4.22 Latch: Mode in which any modifier key remains logically pressed (active) in combination with a single subsequent non-modifier keypress or pointing device button action.

NOTE: Contrast with **Lock** (4.23).

4.23 Lock: Persistent mode in which one or more modifier keys or pointing device buttons remain logically pressed (active) until lock mode for the key or button is turned off.

NOTE 1: Contrast with latch (4.22): unlike LATCH which affects only keyboard and pointing device actions, LOCK would affect any software that uses the modifier key(s) to alter its behavior.

NOTE 2: Lock mode is usually turned off explicitly by the user - but may also be turned off at other times such as system shutdown or restart.

4.24 Modifier Key: Keyboard key that changes the action or effect of another key or a pointing device

NOTE: Modifier key states are sometimes used to alter software behavior in other ways as well such as when inserting a CD to prevent the operating system from automatically playing the CD

EXAMPLE 1: Moving the keyboard focus with the shift key held down extends the current selection in the direction of cursor movement, rather than merely moving the position of the cursor..

EXAMPLE 2: Pressing “C” results in the input of that character, and pressing “Ctrl+C” results in a “Copy” function.

4.25 Name: word or phrase that is associated with a user interface element and is used to identify the element to the user

NOTE 1: Names are most useful when they are the primary word or phrase by which the on screen instructions, software documentation, and its users refer to the element, and do not contain the type or status of the user interface element.

NOTE 2: Contrast with label (4.17). The Name attribute may or may not be presented to users but is available to assistive technologies. It contrasts with label which, In this part of ISO 9241, refers to the presented title for a user interface element. Textual labels are often a visual display of the name.

NOTE 3: When a textual label is provided it would generally present the name or a shortened version of the name. Not all user interface elements have labels however. In those cases the names would be available to assistive technologies (or sometimes by pop up tool tips, etc.).

NOTE 4: Names should not be confused with an internal identifier (ID) which may be used by software and which may not be designed to be understood by a human.

EXAMPLE 3: A window displays the image of a printer that the user can click to print the current document. This image’s label is the image, its name attribute

is “Print”, its role attribute is “Push Button”, and its description attribute is “A printer”.

4.26 Natural Language: Language which is or was in active use in a community of people, and the rules of which are mainly deduced from the usage

4.27 Platform software: software that interacts with hardware or provides services for other software

NOTE 1: A browser can function both as an application and as platform software.

NOTE 2: In this document the use of the word software by itself means both platform software and application software.

EXAMPLES: An operating system, device drivers, windowing systems, and software toolkits.

4.28 Pointer: Graphical symbol that is moved on the screen according to operations with a pointing device

NOTE 1: The location or representation of the pointer may also change to reflect the current state of software operations.

NOTE 2: Users typically interact with user interface elements on the screen by moving the pointer to an object's location and manipulating that object.

NOTE 3: Examples of devices that are used to control pointers include mice, tablets, fingers, and 3D wands. The pointer can also be moved using the keyboard (e.g. MouseKeys).

NOTE 4: Although the pointer is sometimes called a “pointing cursor”, this document uses the word ‘cursor’ only to refer to an indicator of keyboard focus.

NOTE 5: Contrast with **Focus cursor** (4.12) and **Cursor** (4.9).

4.29 Pointer Focus: current assignment of the input from the pointing device to a window

NOTE The window with pointer focus usually has some distinguishing characteristic, such as a highlighted border and/or title bar.

4.30 Pointing Device: Hardware and associated software used to position the pointer

NOTE 1: Pointing devices usually have buttons that are used to activate or manipulate user interface elements.

NOTE 2: Almost any hardware can be used to control the pointer with appropriate software.

EXAMPLES: Pointing devices include mice and trackballs, but they may also include head trackers, hand or finger trackers, touch screens, tablet pens, voice input, and many other hardware/software combinations that systems treat as a pointing device.

4.31 Screen Reader: Assistive technology that allows users to operate software without needing to view the visual display

NOTE 1: Output of screen readers is typically text-to speech or dynamic Braille output on a refreshable Braille display.

NOTE 2: Screen readers rely on the availability of information from the platform and applications, such as the name or label of the user interface.

4.32 Text cursor: visual indication of the current insertion point for text entry

NOTE Contrast with “pointer” and “focus cursor.”

4.33 Usability: Extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use

4.34 User Interface Element: A user interface entity that accepts input, provides information, and/or groups other user interface elements

NOTE 1: User interface elements may or may not be interactive.

NOTE 2: Both entities relevant to the task and entities of the user interface are regarded as user interface elements. Different user interface element types are text, graphics and controls. A user interface element may be a visual representation or an interaction mechanism for a task object (such as a letter, a sales order, electronic parts, or a wiring diagram) or a system object (such as a printer, hard disk, or network connection). It may be possible for the user to directly manipulate some of these user interface elements.

NOTE 3: User interface elements in a graphical user interface include such things as basic objects (such as window title bars, menu items, push buttons, image maps, and editable text fields) or containers (such as windows, grouping

boxes, menu bars, menus, groups of mutually-exclusive option buttons, and compound images that are made up of several smaller images). User interface elements in an audio user interface include such things as menus, menu items, messages, and action prompts.

NOTE 4: Also referred to as “user interface object” as in HFES 200.3

5 Rationale and Benefits of Implementing Accessibility

Accessibility is an important consideration in the design of products, systems, environments and facilities because it affects the range of people who are able to use them.

Accessibility can be improved by incorporating features and attributes known to benefit users with special requirements. To determine the achieved level of accessibility, it is necessary to measure the effectiveness, efficiency, and satisfaction of users working with a product or interacting with an environment for the widest range of users.. Measurement of accessibility is particularly important in view of the complexity of the interactions with the user, the goals, the task characteristics and the other elements of the context of use. A product, system, environment or facility can have significantly different levels of accessibility when used in different contexts

Planning for accessibility as an integral part of the design and development process involves the systematic identification of requirements for accessibility, including accessibility measurements and verification criteria within the context of use. These provide design targets that may be the basis for verification of the resulting design.

The approach adopted in this part of HFES 200 has the following benefits.

- The framework can be used to identify the aspects of accessibility and the components of the context of use to be taken into account when specifying, designing or evaluating the accessibility of a product.
- The performance and satisfaction of the users can be used to measure the extent to which a product, system, environment or facility is accessible in a specific context.
- Measures of the performance and satisfaction of the users can provide a basis for determining and comparing the accessibility of products having different technical characteristics, which are used in the same context.
- The accessibility planned for a product can be defined, documented and verified (e.g. as part of a quality plan).

6 Sources of Variation in User Characteristics

© Copyright 2000–2008, Human Factors and Ergonomics Society. All rights reserved. No portion of this document may be reproduced, stored, or transmitted by any means without prior written permission of the Human Factors and Ergonomics Society.

HFES 200.2

All user populations vary significantly in terms of their characteristics, capabilities and preferences. Any interactive system will include, within the user group for which it is designed, people with very different physical, sensory and cognitive abilities. These differences will have many different sources including innate characteristics, culture, experience and learning, as well as changes that occur throughout life. While the recommendations in this standard are based on the current understanding of the individual characteristics of people who have particular physical, sensory and cognitive impairments, their application addresses the diversity of abilities within any intended user population that may lead to limitation on activities.

Disabilities considered include not only those due to restrictions on mobility or physical performance, such as loss of a limb or tremor, but also those associated with sensory impairment, such as low vision or hearing loss, as well as cognitive factors, such as declining short term memory or dyslexia. Annex C provides an outline of some of the limitations typically encountered by individuals with various types of disability, but does not constitute an exhaustive account of all the issues that may arise. In addition, limitations on activities may be created by the environment, such as stress, poor lighting or noise, or by the need to deal with other tasks in parallel, such as taking care of another person, and these are also taken into account.

The extent to which the particular source of any disability creates limitations varies and some of the guidance provided will be specific to the degree of disability experienced. Thus visual impairments may range from declining ability to resolve small detail to having been blind from birth. Different provisions in the design of the interactive system may be needed to deal with different degrees of disability. For example the facility to enlarge the size of detail presented on a screen will not address the problems of those people who are blind. It is also important to recognize that people may experience multiple forms of disability. Guidance that is appropriate to addressing a specific type of disability may not work if somebody who has that disability also has some other type of disability. For example, auditory output of written text will not provide support for the deaf-blind. It is therefore important that different approaches to access be supported so that interfaces can be individualized to the user and their task.

7 How to Use This Standard

7.1 General

Users of this standard will need to consider all the recommendations that it contains. In many cases the application of a recommendation may depend on the particular context of use so that it will be necessary to determine whether a given piece of guidance is applicable.

In order to achieve accessibility it is necessary to provide support in different parts of the system of software, which includes platform software (the operating system and

associated layers, and toolkits) and other software (such as most applications) that run on and make use of services provided by platform software.

While much can be done to improve accessibility in the design of an application, it is not possible to provide all of the input and output support that users require in every circumstance at the application level alone. To the extent that any particular part of the software is dependent upon a level below it for its operational characteristics it will be necessary to ensure that the lower levels enable the implementation of recommended accessibility characteristics in any layers that depend upon them. Similarly, accessibility characteristics implemented by the platform may require cooperation from layers running on top of them in order to be fully effective. The majority of the requirements and recommendations in clauses 8, 9, 10 and 11 require that the issue be addressed at more than one level of the software system if the particular requirement or recommendation is to be satisfied.

These dependencies may occur in relation to different layers in the platform (e.g. window management on top of process management and screen drawing, which are on top of hardware drivers) and in relation to the applications that are mounted on the platform. Applications themselves may have layers that result in dependencies arising within different levels of the application.

Most of the guidance in this document applies to all software that implements or contributes to the software user interface, regardless of whether or not it is part of the platform. Some guidance is only applicable to portions of platform software (such as guidelines about low-level input, window management, or system-wide behaviours); for example, the platform is the general means by which accessibility features that involve control of hardware devices, in particular those involving input, are implemented and controlled. Similarly, other guidelines may only apply to software that displays user interface elements, generates sounds, or exhibits other specific behaviours. In these cases the applicable layers or type of software is indicated in the text of the requirement or recommendation, and may further be elaborated upon in notes accompanying the guidelines.

7.2 Conformance

If a product is claimed to have met the applicable recommendations in any part of HFES 200, the procedures used in establishing requirements for, developing, and/or evaluating the recommendations shall be specified. The level of specification of the procedure is a matter of negotiation between the involved parties.

Software used on, or intended to be used on closed systems should be evaluated in conjunction with the intended hardware configuration and conform to all clauses except section 8.5.

© Copyright 2000–2008, Human Factors and Ergonomics Society. All rights reserved. No portion of this document may be reproduced, stored, or transmitted by any means without prior written permission of the Human Factors and Ergonomics Society.

Server software (used in client-server and mainframe environments) should be evaluated in conjunction with the client (including terminal) software that will be used with it.

In part 2 (only) of this standard the provisions are assigned to three levels in order to provide information on the relative priority of the provisions. This also provides harmonization with ISO 9241-171, which also assigns priority levels to provisions. ISO 9241-171 has two levels of provisions (SHALL and SHOULD). HFES 200.2 has three levels. Level 1 provisions in HFES 200.2 correspond to 9241-171 SHALL statements. Levels 2 and 3 provisions of HFES 200.2 correspond to 9241-171 SHOULD statements. Three (rather than two) levels were provided in HFES 200.2 to provide additional prioritization information for the large number of provisions beyond Level 1. In addition, some provisions were designated as level 3 provisions because they may be difficult or inappropriate to implement in some circumstances. It should be noted that provisions at all three levels are important to different individuals and that these provisions do not constitute all that can be done to make software more accessible to people with all types, degrees and combinations of disability.

8 General Guidelines

8.1 Names and labels for user interface elements

8.1.1 Provide a name for each user interface element (Level 1)

Software should associate an identifying name with every user interface element except where the name would be redundant.

NOTE 1: A name conveys the identity of the user interface element to the user. It complements the role attribute that tells an element's function (such as that it acts as a push button) and the description attribute that summarizes the element's visual appearance.

EXAMPLE 1: The application provides a label showing the name "File name" for a static text field that shows the name of the file being described in the fields below.

EXAMPLE 2: Dialog boxes or windows have meaningful names, so that a user who is hearing rather than seeing the screen gets appropriate contextual information.

NOTE 2: If some names are missing, assistive technology may be unable to sufficiently identify or operate the user interface elements for the user.

NOTE 3: Names would be redundant for user interface elements whose entire informational content is already conveyed by their role attribute (such as a horizontal rule), static text elements that serve to name other elements, and elements that only serve as an integral portion of a parent element (such as the rectangular border around a button).

EXAMPLE 1: The software does not need to provide a name for a static text field that says "Last name:" and serves to identify text box that follows it, as that string would be exposed using the field's value attribute.

EXAMPLE 2: The software does not need to provide a name for an image that forms part of a raised border around a non-standard button.

NOTE 3: In some cases the name will be displayed visibly, but in other cases it will only be provided programmatically for use by assistive technology as described in section 8.5.

EXAMPLE 1: A control is listed in the product documentation as "the Print button," therefore its identifying name in the software is "Print" (regardless of whether the word "Print" appears on the visual representation of the button or not).

NOTE 4: Such elements may be containers that serve to group one or more sub-elements. In a typical graphical user interface, examples of UI elements include basic elements such as window title bars, menu items, push buttons, images, text labels and editable text fields, while examples of containers include windows, list boxes, grouping boxes, menu bars, menus, groups of mutually-exclusive option buttons, and compound images that are made up of several smaller images. In a typical audio user interface, examples of interactive UI elements include menus, menu items, messages, prompt tones, and pauses.

EXAMPLE 1: Compound user interface elements that consist of a collection of other user interface elements have a group name. A Web page image composed of a series of smaller image files provides a group name ("Construction Site") in addition to the names of the individual component images ("building", "bulldozer", "dump truck", "crane", etc.).

8.1.2 Provide meaningful names (Level 2)

Names of user interface elements should be comprised of natural language words that are meaningful to the intended users.

NOTE 1: This means that each word in the name would occur in a standard dictionary or in electronic documentation for end-users included with the software.

NOTE 2: The names are most useful when they are the primary name by which the software, its documentation, and its users refer to the element and do not contain the type or status of the user interface element.

EXAMPLE 1: The name of a checkbox is 'Gender' and not 'Gender checkbox.'

NOTE 3: User interface elements that represent a real entity (such as a documents, location, or person) may be provided with the name of that entity even if the name is too lengthy or cryptic to be read easily.

NOTE 4: Names may use terms that are specific to a particular task domain provided that they have established meanings for the intended users.

EXAMPLE 1: A control is listed in the product documentation as “the Print button,” therefore its identifying name in the software is “Print” (regardless of whether the word “Print” appears on the visual representation of the button or not).

EXAMPLE 2: Dialog boxes or windows have meaningful names, so that a user who is using speech output because they cannot see the screen gets appropriate contextual information

8.1.3 Provide unique names within context (Level 3)

Each name of a user interface element specified by software developers should be unique within its context.

NOTE 1: Users will not be able to use the name to identify an element if several elements have the same name within the same context.

NOTE 2: A name is considered unique if no other user interface element with the same name and role attributes shares the same container or parent element (such as a window, group box, section, etc.)

NOTE 3: User interface elements that represent a real entity (such as a documents, location, or person) may be provided with the name of that entity even if the name is not unique in its context.

EXAMPLE 1: A form displays areas containing fields with a customer's home and business details. Each area has an associated "Change" button. Rather than duplicate the name, the buttons are named "Change Home" and "Change Business

EXAMPLE 2: A form for a purchasing application has several rows of items, each containing a text field displaying the title of a book, followed by "Buy" button that is used to purchase that book. Even though the face of each button looks identical, the

form is implemented so that each provides a unique name for use by assistive technology, such as "Buy The Grapes of Wrath" and "Buy Pride and Prejudice".

EXAMPLE 3: A user opens a second window of the same document using their word processing application. Both windows are of the same document and both are editable. The word processor adds a ":1" to the end of the document name to form the name of the first window. It names the second window with the same document name except that it appends a ":2" to the end of the second window name so the two windows have unique names.

EXAMPLE 4: When a script or object hosted within a Web browser attempts to set the window's title to a string that is already used by another of the browser's windows, the browser modifies this string to be unique.

8.1.4 Make names available to assistive technology (Level 1)

Except when running on closed systems, each name of a user interface element and its association should be made available by the software system to assistive technology in a documented and stable fashion

NOTE 1: Section 8.5.4 ("Make user interface element information available to assistive technologies") describes how to make the information available.

NOTE 2: In a platform that does not provide a standard service for the association of names and elements, application developers document how assistive technologies can access that information.

8.1.5 Display names (Level 2)

If a user interface element has a visual representation and is not part of standard components of the user interface, software should present its name to users (either by default or at the user's request).

NOTE: Standard components of the user interface are components that are provided by the platform and look and behave the same across applications.

EXAMPLE 1: In an application the window scroll up and scroll down buttons do not have a label or pop up text because these buttons are standard across applications on the platform. At the bottom of the scrollbar however are special "jump to next find" and "jump to last find" arrows that do have pop up text that describe their function.

EXAMPLE 2: A print button has a picture of a printer with a textual name that pops-up when the user pauses a pointer over the button and also when the user moves the keyboard focus to the button and presses a specific keyboard command.

8.1.6 Provide names and labels that are short (Level 2)

Each name or label of a user interface element specified by software developers should be short enough to be rendered succinctly..

NOTE 1: Developers are encouraged to put the most distinctive parts of the name first, so users can skip over the latter parts once they have read enough to identify the element.

NOTE 2: User interface elements that represent a real entity (such as a document's, location, or person) may be provided with the name of that entity even if the name is too lengthy or cryptic to be read easily.

NOTE 3: Using brief labels also benefits users of auditory, visual and tactile output.

EXAMPLE 1: "Print" is used instead of "Print button" or "This button prints the current document."

EXAMPLE 2: An icon representing a document is labeled with the file name or title of the document, even if that string is too lengthy or cryptic to be read easily, because that string was determined by the document author rather than by software developers.

8.1.7 Provide text label display option for icons (Level 3)

Software should allow users to choose between displaying: a) icon images with text labels, b) icon images only, or c) icon text labels only..

NOTE 1: It is useful for the user to be able to adjust the font size

8.1.8 Properly position the labels of user interface elements on screen (Level 3)

The labels for user interface elements provided by software should be consistently positioned, relative to the elements that they are labeling, on the display (HFES 200.3 Section 8.3.4.).

If the platform software has conventions for positioning labels relative to the elements that they are labeling, these conventions should be followed.

NOTE 1: This helps assistive technology correctly associate the labels with their corresponding controls, and helps users of screen magnification software know where to look for a label or control.

8.2 User preference settings

8.2.1 Enable individualization of user preference settings (Level 2)

© Copyright 2000–2008, Human Factors and Ergonomics Society. All rights reserved. No portion of this document may be reproduced, stored, or transmitted by any means without prior written permission of the Human Factors and Ergonomics Society.

When the software enables the user to set personal preferences, these settings should be easily adjustable..

EXAMPLE 1: A software application allows users to configure and save settings for font size and style within a particular window.

NOTE 1 It is very important to use system-wide user preference settings provided by the platform, in addition to any preference settings for product-specific options

EXAMPLE 2: A software application allows a user with cognitive disabilities to choose the number and size of icons displayed at any one time.

NOTE 2: Requiring users to hand edit a configuration file is not an easy method for individualizing preference settings because it is too easy for the user to accidentally enter invalid values or otherwise corrupt the file.

EXAMPLE 3: A user chooses preference settings through a graphical user interface, rather than directly editing the configuration files.

NOTE 3 Business considerations related to consistency of operations, performance-based considerations, safety, privacy, and security concerns may all lead to some necessary restriction by system administrators of the user's capability to modify the behavior and appearance user interface elements in certain contexts. Administrators need to show restraint in limiting user control. Not all options/preferences settings are appropriate for such administrative control.

NOTE 4 Administrators within this business environment can make specific permission profiles for users which require more flexibility in their options/preference settings for usability and accessibility.

8.2.2 Enable adjustment of attributes of common user interface elements (Level 2)

Software should enable users to adjust the attributes of common user interface elements, if applicable to the task.

NOTE 1: "Common attributes" for a visual interface could include, but are not limited to, font type, font size, and font color. For an auditory interface they could include, but are not limited to, aural cue type, rate, volume, pitch, position in 3D audio space, etc. For a tactile interface they could include, but are not limited to, haptic object size, texture, xy- or xyz-position, pressure sensitivity, solidity, etc."

NOTE 2: Platform software often supports these options for standard user interface elements it provides. To enhance user experience, applications can use the settings defined at the platform level.

EXAMPLE Software retains user preference for window size and location between sessions.

8.2.3 Enable individualization of the user interface look and feel (Level 3)

Software should provide a mechanism enabling users to individualize the interface look and feel including the modification or hiding of command buttons.

EXAMPLE 1: A user with a cognitive disability may, when using a given application, change the interface to simplify the application's look and feel.

EXAMPLE 2: A word processor allows users to temporarily hide menu items and tool bar buttons that they do not find useful for a given situation.

8.2.4 Enable individualization of the cursor and pointer (Level 1)

If the hardware supports the service, software should enable users to individualize attributes of all keyboard focus cursors, text cursors, and pointers including but not limited to shape, size, stroke width, color, blink rate (if any) and pointer trails (if any).

NOTE 1: Platform software often supports these options for standard cursors and pointers it provides, and software using these cursors and pointers may automatically comply with this guideline.

NOTE 2: The ability to set the cursor to non-blinking is important for users with attention deficits, who may be easily distracted.

NOTE 3: The color aspect of this provision is not applicable if the presentation of the cursor or pointer is an inversion of the image and it has no color.

EXAMPLE 1: Users with low vision can change a text cursor from non-blinking to blinking, and adjust the size to be more readily visible given their visual capabilities.

EXAMPLE 2: Users with low vision and/or a color deficiency can change the thickness and color of the keyboard focus cursor so that they can more easily see the current input focus.

EXAMPLE 3: Users with low vision can make the pointer larger so that they can more readily locate it.

8.2.5 Provide user-preference profiles (Level 3)

Software should enable users to create, save, edit and recall profiles of preference settings, including input and output characteristics, without having to carry out any restart that would cause a change of state or data.

© Copyright 2000–2008, Human Factors and Ergonomics Society. All rights reserved. No portion of this document may be reproduced, stored, or transmitted by any means without prior written permission of the Human Factors and Ergonomics Society.

HFES 200.2

NOTE 1: For systems that provide access for multiple users, such as library systems, conversion back to a default profile may be advisable.

NOTE 2: It is often useful to be able to access the preference settings over a network. Doing this in a secure way would preserve privacy especially for people who are worried about revealing the fact that they have a disability

NOTE 3: It is advisable to minimize the need to restart the system or application in order for changes in user interface settings to become effective.

EXAMPLE 1: Platform software allows each user to save global settings for font size, sound volume, and pointer-control settings that apply everywhere on the system.

EXAMPLE 2: The profile for a public library system is modified for the needs of a current user but returns to default values when that user is finished.

EXAMPLE 3: A user is able to load a system default configuration quickly on a computer that is currently using an alternative configuration for a user who has a disability.

EXAMPLE 4: A user is completing an on-line process and has to make adjustments to the accessibility feature to reduce errors. Restarting the operating system or the user agent would cause them to lose their work.

8.2.6 Provide capability to use preference settings across locations (Level 3)

Software should permit users to transfer their preference settings easily onto a compatible system.

NOTE 1: Portability is important for users with disabilities because they may find a system difficult or impossible to use without the preferences set to meet their interaction needs. The overhead and effort required to create preference settings can be a significant hindrance to system usability if it must be repeated at every location.

NOTE 2: User preferences profiles are sometimes made publicly available e.g. for download from internet. Because people can be concerned about others knowing of their disability, it would be helpful if their use of these resources could be kept private.

NOTE 3: Some platform software may provide a general mechanism for transferring their preference settings; in such cases software may not have to implement this feature itself, as long as it follows platform conventions for storing its user preference settings.

EXAMPLE 1: A user visiting a different building on the company network logs in and the system automatically locates and uses his or her personal preference settings from the network without having to edit configuration files.

EXAMPLE 2: A user loads a preference settings file from a USB drive onto a new computer.

EXAMPLE 3: A user's preference settings are loaded from a smart card onto a new system.

8.2.7 Enable user control of timed responses (Level 1)

Unless limits placed on the timing of user responses are essential to maintaining the integrity of the task or activity or are based on real life time constraints (e.g. an auction), software should allow users to adjust each software-specified user response time parameter in one or more of the following ways:

- 1) the user is allowed to deactivate the time-out, or;
- 2) the user is allowed to adjust the time-out over a wide range which is at least ten times the length of the default setting, or;
- 3) the user is warned before time expires, allowed to extend the time-out with a simple action (for example, "hit any key") and given at least 20 seconds to respond.

EXAMPLE: A logon prompt requires the user to enter their password within 30 seconds. Show the remaining time on the screen and provide a control to stop the time decrementing.

8.3 Special considerations for accessibility adjustments

8.3.1 Make controls for accessibility features discoverable and operable (Level 1)

Software should enable any On/Off controls and adjustments for accessibility features to be discoverable, and operable by those who need that feature

NOTE Features are discoverable if their settings and description can be found by browsing the user interface (including browsing Help available through the application).

EXAMPLE 1: Rather than needing to use a combination of keys, a user can turn the "StickyKeys" accessibility feature on and off by pressing the Shift key five times in succession.

EXAMPLE 2: Accessibility features are turned on by use of a single toggle key held down at system start-up.

EXAMPLE 3: Controls for individualizing low-vision options are shown in large type by default.

8.3.2 Safeguard against inadvertent activation or deactivation of accessibility features (Level 2)

Software should prevent inadvertent activation or deactivation of accessibility features.

EXAMPLE 1: The software system requests confirmation before activating or deactivating accessibility features.

8.3.3 Avoid interference with accessibility features (Level 1)

Software should not disable or interfere with the accessibility features of the platform.

EXAMPLE 1: Software which intercepts keyboard input does not defeat the operation of keyboard filters such as key latching and locking.

8.3.4 Inform user of accessibility feature on/off status (Level 2)

Software should allow the user to identify the current status of accessibility features.

EXAMPLE 1: A control panel shows the current state of all accessibility features.

EXAMPLE 2: A small icon on the screen indicates that an accessibility feature is switched on.

8.3.5 Inform user of accessibility feature activation (Level 2)

When an accessibility feature can be activated unintentionally, software should inform users and provide an opportunity to accept or cancel the activation.

NOTE 1: The alert for an individual accessibility feature could be disabled at user request but would be on by default.

NOTE 2 It is good practice to provide an alert whenever SlowKeys is activated via a keyboard shortcut method since an accidental activation could lead an uninformed user to think the keyboard was broken. Regular users of SlowKeys however prefer to have a way to defeat the alert so that they do not have to dismiss the alert each time they turn their SlowKeys feature on.

8.3.6 Enable persistent display (Level 2)

© Copyright 2000–2008, Human Factors and Ergonomics Society. All rights reserved. No portion of this document may be reproduced, stored, or transmitted by any means without prior written permission of the Human Factors and Ergonomics Society.

When users can activate a menu, control, or other user-interface element to display additional information or controls, software should allow that information or control to persist while the user engages in other tasks, until the user chooses to dismiss it, if it is appropriate to the task.

NOTE: Persistent display of frequently used windows and controls may be helpful for users who have physical, language, learning or cognitive disabilities, and reduces the number of steps required to access them.

EXAMPLE 1: Users can keep a Help Window available as they go through the tasks described.

EXAMPLE 2: The user can “tear off” one or more menus and continue to view and/or use them while navigating and using other menus.

EXAMPLE 3: The user can add a toolbar button that duplicates the function of a specific menu command. By doing so this menu command is persistently displayed.

8.4 General control and operation guidelines

8.4.1 Enable switching of input/output alternatives (Level 3)

Platform software should enable users to switch among the available input/output alternatives without requiring them to reconfigure or restart the system or applications, unless there would be no change of state or data.

NOTE: This capability aids users with different abilities who are working together on the same system.

EXAMPLE 1: A person who is blind uses their system only through the keyboard, using keyboard substitutes for mouse actions. A sighted user working on the same system can use the mouse and type in text. The system does not have to be restarted in between each session.

EXAMPLE 2: One user points with a mouse to an icon for a document on their screen and says “print” to print the document while another clicks on the document and types Ctrl+P to print it. Still another uses the ‘Print’ item in the “File” menu to print the document.”

EXAMPLE 3: To implement a setting change, an application must restart, but it restores all data including position of the keyboard focus cursor.

8.4.2 Optimize the number of steps required for any task (Level 3)

© Copyright 2000–2008, Human Factors and Ergonomics Society. All rights reserved. No portion of this document may be reproduced, stored, or transmitted by any means without prior written permission of the Human Factors and Ergonomics Society.

HFES 200.2

Software should be designed to optimize the number of steps that the user has to perform for any given task

NOTE 1: It is important to find a balance between reducing the steps to improve efficiency and adding steps to allow for sufficient explanation of an infrequent task.

EXAMPLE 1: A user who wants to print one document can do so in only two steps. Once they select the Print icon on the toolbar, a dialogue is displayed (which can, if desired, be used to change various print settings) and the user simply selects the OK button.

EXAMPLE 2: A user with cerebral palsy types very slowly, and so finds it much more convenient when they can save a document by pressing a single key combination rather than having to navigate menus and dialog boxes

8.4.3 Provide “Undo” and/or “Confirm” functionality (Level 2)

Software should provide a mechanism that enables users to undo at least the most recent user action and/or cancel the action during a confirmation step

NOTE 1: Although this is a general ergonomic principle, “Undo” mechanisms are particularly important for users who have disabilities that significantly increase the likelihood of an unintentional action. These users can require significant time and effort to recover from such unintentional actions.

NOTE 2: A macro is considered to be one user action

NOTE 3: Generally, the more consecutive actions the user can undo, the better.

NOTE 4: It is preferable if “undo” operations themselves can be undone.

NOTE 5: This may not be possible for such interactions as operations which cause fundamental transformation of logical or physical devices, or may involve a data exchange with 3rd parties that are out of the software’s control, etc.

NOTE 6: It is preferable that the default configuration provides a confirmation step for any actions that the user cannot undo with a single Undo command. Software may allow the user to disable the confirmation for specific actions..

EXAMPLE 1: A user with Parkinson's disease may inadvertently input a sequence of keystrokes, which activate several dialogues that need to be undone. The use of several steps of the undo function may permit the user to go back to the original state.

EXAMPLE 2: A user is about to format a hard disk. As this is an operation that cannot be undone, the software shows a confirmation dialog before the formatting begins.

8.4.4 Provide alternatives when assistive technology is not operable (Level 1)

If a task requires user interaction when the state of the software prevents use of assistive technology or speech output (such as during system start up), the software should provide the user with an alternative means, which does not require user interaction during that period, for completing the task.

NOTE: System start-up and restart include operations prior to the stage where the user's accessibility aids and preference settings are available. They can be either accessible or non-interactive.

EXAMPLE 1: A computer is configured with a small "boot loader" that lets the user to choose between two or more operating systems present on the system. Because the boot loader's menu is run before a full operating system or any assistive technology is running, it provides a mechanism by which the user can, during a normal session and using their assistive technology, specify which operating system will be loaded the next time the system starts.

EXAMPLE 2: A computer does not request any password or other user input until after it has loaded access features.

EXAMPLE 3: A public information kiosk restarts automatically and comes up accessible. There is no log on before access features work.

8.4.5 Enable software-controlled media extraction (Level 1)

If the hardware allows for it, software should enable the user to perform software-controlled media extraction.

NOTE 1: Those media include, but are not limited to, floppy disks, CD-ROMs and DVDs.

NOTE 2: In most cases the user can use this feature as provided by the platform software without explicit support from software applications.

EXAMPLE: A user who cannot press physical buttons on the computer or handle a CD-ROM can nevertheless use their on-screen keyboard to instruct the operating system to eject the disc, so that it will not interfere with the computer's next restart.

8.4.6 Support "Copy" and "Paste" operations (Level 2)

© Copyright 2000–2008, Human Factors and Ergonomics Society. All rights reserved. No portion of this document may be reproduced, stored, or transmitted by any means without prior written permission of the Human Factors and Ergonomics Society.

HFES 200.2