Human Interface Note

Note #2 **Design Principles for On-Line Help Systems**

Written by: Kathleen Gomoll & Anne Nicol January 1990
(Supersedes Human Interface Update #12)

Discussion of a set of criteria and guidelines for on-line help.

Introduction

As part of an ongoing effort to design and support a consistent interface to on-line help for our computers, Apple has been developing a set of criteria and guidelines for on-line help. These criteria are based on observations of users in our lab, reviews of the research, requests and comments from our developers, and—last, but not least—the *Human Interface Guidelines: The Apple Desktop Interface*. This is a working document. It reflects Apple's current view on designing on-line help, and we will probably revise and expand it as we progress. In the future, we intend to distribute specific guidelines regarding access to on-line help and the display of help information. Ultimately, we hope to supply toolbox support for the interface features that we find, through user testing, to be most effective.

This document is divided into three sections: Principles, General guidelines, and Hints for structure and organization. The Principles section reflects Apple's underlying design philosophy for on-line help. The General guidelines section puts guidelines for designing on-line help into the context of the principles outlined in *Human Interface Guidelines*. Finally, the Hints section lists suggestions for organizing and structuring help information. These hints come from developers and from current research.

Principles

On-line help should never be a substitute for good interface design.

This is our first and foremost principle. Before setting out to build a help system that "explains" a difficult interface, try to identify what makes the interface difficult—and fix the problems. When you have made your interface as clear as it can be, then develop a help system that aids users as they work.

Help should be context-sensitive; it should not take the user away from the task at hand.

Perhaps the biggest complaint users have about help systems is that they don't want to leave their current application to get help. When users are forced to leave the context of their problem, they often forget the specifics of the problem. Also, users often have trouble applying the help information once they get back to the application because the help is no longer visible.

Help systems should assist users in framing their questions and provide different types of help for different questions.

When users need help, they often turn to local experts and ask questions. Human experts are often able to help users frame their questions so they can get the appropriate answer. Once the right question has been asked, help can be delivered quickly. Users' questions fall into a relatively small number of distinct categories, and those categories call for different types of assistance. For example, we can make a clear distinction between the question "What is this?" and the question "How do I do this?"

Help systems should be dynamic and responsive to individuals.

Different users need different kinds of help because they have individual learning styles and needs. For example, some users may want to be shown exactly how to do something, while others may want to explore and learn by their mistakes. If possible, on-line help systems should make use of the user's competence, learning style, level of experience and past actions to provide appropriate help.

Users shouldn't need help on how to get help.

Help systems should be structurally simple and self-explanatory. Although your help system may require a few words of instruction (like "click here" or "select a topic"), don't fall into the trap of turning your help system into a complicated application that requires lengthy instructions.

General Guidelines

Make help accessible through recognition, not recall.

See-and-point (instead of remember-and-type): Users can choose any available action at any time—without having to remember a particular command or name. This paradigm requires only recognition, rather than recall, of the desired activities.

Put the help system under the user's control.

Direct manipulation: Users want to feel that they are in charge of the computer's activities. The user, not the computer, initiates and controls all actions. If the user attempts something risky, the computer provides a warning, but allows the action to proceed if the user confirms it. This approach "protects" the beginner but allows the user to remain in control.

Support exploratory behavior by making an interactive help system.

User Control: People learn best when they're actively engaged. Too often, however, the computer acts and the user merely reacts within a limited set of options. Allow users to try things out.

Place help options where they are visible to the user.

Direct manipulation: Users want topics of interest to be highlighted. They want to see what functions are available at any given moment.

See-and-point: Users rely on recognition, not recall; they shouldn't have to remember anything the computer already knows.

Use graphics, animation, and sound.

Principles of Graphic Communication: The real point of graphic design, which comprises both pictures and text, is clear communication. In the Apple Desktop Interface, everything the user sees and manipulates on the screen is graphic.

Metaphors from the real world: Whenever appropriate, use audio and visual effects that support a real-world metaphor. Use animation for modeling user actions. Use sound for orienting attention and reinforcing information.

Hints for Structure and Content

On-line help should not be simply an on-line version of the print documentation.

As a method for communication, computers provide opportunities that books can't provide. Use the computer's capacity to its fullest by designing a help system that brings help to the user rather than requiring the user to page through an on-line book. Use the computer to link information in useful ways, and to create graphics, sound, animation, and examples.

Organize the help system in very small, addressable chunks of information.

By creating a help system that is modular, you allow tremendous flexibility. Small chunks of information can be grouped in strategic ways to provide users with only the most relevant information.

Include both search and browse capabilities.

Build a “find” feature into your help system to allow users to quickly search for specific topics. Also allow users to browse through available help topics, since it’s often easier to recognize a topic than to think of an appropriate keyword.

Allow users to discard help.

Users should never be forced to use the help system to use an application. A help system should always be an optional aid.

Make the help system customizable and editable.

To make the most efficient use of a help system, users should be able to customize and edit the information to suit their own needs. For example, users may want to put a marker on a piece of information they access frequently, or they may want to eliminate or change information that doesn’t suit their needs.

Include help information that can be delivered automatically.

Sometimes users make the same error repeatedly. Rather than waiting for the user to ask for help, the help system should be able to detect problems and offer help automatically.

Incorporate hypertext features for linking information.

Hypertext allows users to press “buttons” to receive context-sensitive help in as much or as little depth as they require. By linking chunks of help information in logical ways, you can develop a help system that is responsive to users’ immediate needs.

Use the help system to inform users about short-cuts.

Short-cuts are facts that experts typically know. A help system that volunteers answers without forcing users to ask questions can help novices become experts.

Suggested Readings

Apple Computer, Inc. (1987). Human Interface Guidelines: *The Apple Desktop Interface*. Reading, MA: Addison-Wesley Publishing Co.

Borenstein, N.S. (1985). *The Design and Evaluation of On-line Help Systems*. Ph.d thesis, Carnegie-Mellon University.

Christensen, M. (1984). *Background for the Design of an Expert Consulting System for On-line Help*. Thesis proposal, Temple University.

Owen, D. (1986). *Answers first, then questions*. In D.A. Norman & S.W. Draper (Eds.), User Centered System Design, (p. 361-375). Hillsdale, NJ: Erlbaum.