Designing Menu Selection Systems

Ben Shneiderman
Department of Computer Science, University of Maryland, College Park, MD 20742

Menu selection systems reduce training and memorization, simplify entry of choices, and structure the user's tasks. However, the use of menu selection is no guarantee that novices or experts will be satisfied or able to carry out their work. This article focuses on the multiple design issues in creating successful menu selection systems. These include the primary issue of semantic organization and the host of secondary issues such as response time and display rates, shortcuts for frequent users, titles, phrasing of menu items, graphic layout, and selection mechanisms. Novel approaches such as pop-up menus and embedded menus are covered. Experimental results and design guidelines are presented.

1. Introduction

Menu selection systems are attractive because they can eliminate training and memorization of complex command sequences. When the menu items are written using familiar terminology, users can select an item easily and indicate their choice either with one or two keypresses or through the use of a pointing device. This simplified interaction style reduces the possibility of keying errors and structures the task to guide the novice and intermittent user. With careful design and high speed interaction, menu selection can become appealing to expert frequent users as well.

Interaction by menu selection is often contrasted with interaction by command language, but the distinctions are sometimes blurred. Typically menu selection requires a single keystroke, whereas commands may be lengthy. However, how would you classify a menu in which the user has to type a six or eight letter item? Typically, menu selection presents the choices on the display, whereas commands must be memorized. However, how would you classify a menu that offered 4 numbered choices and accepted 10 more generic choices that are not displayed? How would you classify a system that offers single letter prompts? What about graphical, two-dimensional menus where selection is made by pointing with a mouse or on a touchscreen? Finally, what category is voice synthesis/recognition menu interaction?

Rather than debate terminology, it is more useful to maintain an awareness of how much the system offers on the display at the moment the selection is made, the form and content of item selection, and what problem domain knowledge is necessary for users to succeed. Menu selection is especially effective when users have little training, are intermittent in using the system, are unfamiliar with the terminology, and need help in structuring their decision-making process.

However, if a designer employs menu selection, it does not guarantee that the system will be appealing and easy to use. Effective menu selection systems emerge only after careful consideration and testing of numerous design issues such as semantic organization, menu system structure, the number and sequence of menu items, titling, prompting format, graphic layout and design, phrasing of menu items, display rates, response time, shortcuts through the menus for knowledgeable frequent users, availability of help, and the selection mechanism (keyboard, pointing devices, touchscreen, voice, etc.).

2. Semantic Organization

The primary task for menu designers is to create a sensible, comprehensible, memorable, and convenient semantic organization. Some lessons can be learned by studying the semantic decomposition of a book into chapters, a program into modules, the animal kingdom into species, or a Sears catalog into sections. Hierarchical decompositions, natural and comprehensible to most people, are appealing because every item belongs to a single category. Unfortunately, in some applications an item may be difficult to classify as belonging to one category, and the temptation to duplicate entries or create a network increases. Despite some limitations, the elegance of tree structures should be appreciated.

Restaurant menus separate appetizers, soups, main dishes, desserts, and drinks to help customers organize

Received April 10, 1985; revised May 10, 1985; accepted June 20, 1985.

their selections. Menu items should fit logically into categories and have readily understood meanings. Restaurateurs who list dishes with idiosyncratic names such as "Veal Monique," generic terms such as "house dressing," or unfamiliar jargon such as "Wor Shu Op" should expect that waiters will spend ample time explaining the alternatives or should anticipate that customers will be anxious because of their insecurity in ordering.

Similarly, for computer menu selection systems, the categories should be comprehensible and distinctive so that the users are confident in making their selections. Users should have a clear idea of what will happen when they make a choice. Computer menu selection systems are more difficult to design than restaurant menus because computer screens typically allow less information to be displayed than do printed menus. Screen space is a scarce resource. In addition, the number of choices and the complexity is greater in many computer applications, and the computer user may not have a helpful waiter to turn to for an explanation.

The importance of meaningful organization of menu items was demonstrated in a study with 48 novice users [1]. Simple menu trees with 3 levels and 16 target items were constructed in meaningfully organized and disorganized forms. Error rates were nearly halved, and user think time (time from menu presentation to user's selection of an item) was reduced for the meaningfully organized form. In a later menu search study, McDonald et al. [2] found that semantically meaningful categories, such as food, animals, minerals, and cities, led to shorter response times than did random or alphabetic organizations. This experiment tested 109 novice users who worked through 10 blocks of 26 trials. The authors conclude that "these results demonstrate the superiority of a categorical menu organization over a pure alphabetical organization, particularly when there is some uncertainty about the terms." With larger menu structures the effect is even more dramatic, as has been demonstrated by studies with extensive videotext databases [3,4].

These results and the syntactic/semantic model [5] suggest that the key to menu structure design is to first consider the semantic organization. The number of items on the screen becomes a secondary issue.

Menu selection applications range from trivial choices between two items to complex videotex systems with 300,000 screens. The simplest applications consist of a *single menu*, but even with this limitation there are many variations. The second group of applications includes a *linear sequence* of menu selections; the progression of menus is independent of the user's choice. Strict *tree structures* make up the third group, which is the most common situation. *Acyclic* (these are menus that are reachable by more than one path) and *cyclic* (there are meaningful paths that allow users to repeat menus) *networks* constitute the fourth group. These groupings describe the semantic organization; special traversal commands may enable users to jump around the branches of a tree, to go back to the previous menu, or to go to the beginning of a linear sequence.

A. *Single Menus*

In some situations, a single menu is sufficient to accomplish a task. Single menus may have two or more items, may require two or more screens, or may allow multiple selections. Single menus may pop up on the current work area or may be permanently available (in a separate window or on a data tablet) while the main display is changed. Different guidelines apply for each situation.

Binary Menus The simplest menu is a binary menu with yes/no or true/false choices such as is found in many home computer games:

DO YOU WANT INSTRUCTIONS (Y,N)?

or in a medical-history taking interview

YOU HAVE HAD SURGERY TO REMOVE YOUR APPENDIX
1 — TRUE
2 — FALSE
MAKE YOUR SELECTION NOW:

Even these simple examples can be improved. In the first case, a novice user might not understand the (Y,N) prompt—really an abbreviated form of the menu of choices. Secondly this common query leaves the user without a clear sense of what is going to happen next. Typing "Y" might produce many pages of instructions and the user might not know how to stop a lengthy output. Typing "N" is also anxiety producing because the user has no idea of what the program will do. Even in writing simple menus, clear and specific choices should be offered which give the user the sense of control:

Your choices are:
1 — Get 12 lines of brief instructions
2 — Get 89 lines of complete instructions
3 — Go on to playing the game
Type a number and press RETURN:

Since this version has three items, it is no longer a binary menu. It offers more specific items so the user knows what to expect, but it still has the problem that users must take instructions now or never. Another strategy might be:

At any time you may type
? — to get 12 lines of brief instructions
?? — to get 89 lines of complete instructions
Be sure to press RETURN after every command
Ready for game playing commands:

This example calls attention to the sometimes narrow distinction between commands and menu selection: the menu choices have become more command-like since the user must now recall the ? or ?? syntax.

The following examples illustrate additional issues in binary menus. Menu items can be identified by single letter mnemonics, as in this photo library retrieval system:

Photos are indexed by film type
B Black and white
C Color
Type the letter of your choice
and press RETURN:

or in a shop-by-computer service:

PLEASE SELECT THE DESIRED METHOD
OF BILLING:
MASTERCARD 1
VISA 2
KEY IN YOUR SELECTION AND PRESS ENTER:

The mnemonic letters in the photo menu are often preferred to the numbered choices in the shop-by-computer menu (see Section 7). The long line of periods can be a distraction, and the uppercase-only lettering may slow reading. The mnemonic letter approach requires additional caution in avoiding collision and increases the effort of translation to foreign languages, but their clarity and memorability are an advantage in many applications. These simple examples demonstrate alternative ways to identify menu items and convey instructions to the user. No optimal format for menus has emerged, but consistency across menus in a system is extremely important.

Multiple Item Menus Single menus may have more than two items. Examples include online quizzes:

Who invented the telephone?
1 Thomas Edison
2 Alexander Graham Bell
3 Lee De Forest
4 George Westinghouse
Type the number and press RETURN:

or the list of options in a document processing system:

EXAMINE, PRINT, DROP, OR HOLD?

The quiz example has distinctive, comprehensible items, but the document processing example shows an implied menu selection that could be confusing to novice users. There are no explicit instructions, and it is not apparent that single letter abbreviations are acceptable. Knowledgeable and frequent users may prefer this short form of a menu selection, usually called a prompt, for its speed and simplicity.

Extended Menus Sometimes the list of menu items may require more than one screen but allow only one meaningful item to be chosen. One resolution is to create a tree-structured menu, but sometimes the desire to keep the system to one conceptual menu is very appealing. The first portion of the menu is displayed with an additional menu item that leads to the next screen in the extended menu sequence. A typical application is in word processing systems, where common choices are displayed first but infrequent or advanced features are kept on the second screen:

SUPERDUPERWRITER MAIN MENU
PAGE 1

1 Edit a file
2 Copy a file
3 Create a file
4 Erase a file
5 Print a file
6 View the directory

P2 Go to PAGE 2

Type the number of your choice
and Press RETURN

SUPERDUPERWRITER MAIN MENU
PAGE 2

7 Alter line width
8 Change character set
9 Attempt recovery of damaged file
10 Reconstruct erased file
11 Set cursor blink rate
12 Set beep volume
13 Run diagnostics

P1 Go back to PAGE 1

Type the number of your choice
and Press RETURN

Sometimes the extended screen menu will go on for many screens of command items or data items. More elaborate scrolling capabilities may be needed.

Pop-up Menus The term "pop-up" or "pull down" menus refers to the process of forcing a menu to appear on the screen in response to a click with a pointing device such as a mouse. The Xerox Star, Apple Lisa, and Apple Macintosh (Figure 1) made these possibilities widely available. There is a great satisfaction on the part of most users in making the menu appear rapidly. Selection can be made by moving the pointing device over the menu items, which respond by highlighting (inverse video, a box surrounding the item, or color have been used).

The contents of the pop-up menu may depend on the position of the cursor when the pointing device is clicked. Since the pop-up menu covers a portion of the screen, there is a strong motivation to keep the menu text small. Hierarchical sequences are also used in pop-up menus.

Permanent Menus Since menus can be used for permanently available commands that can be applied to a displayed object. For example, the Bank Street Writer, a word processor designed for children, always shows a fragment of the text and this menu:

ERASE	MOVE	FIND	TRANSFER
UNERASE	MOVEBACK	REPLACE	MENU

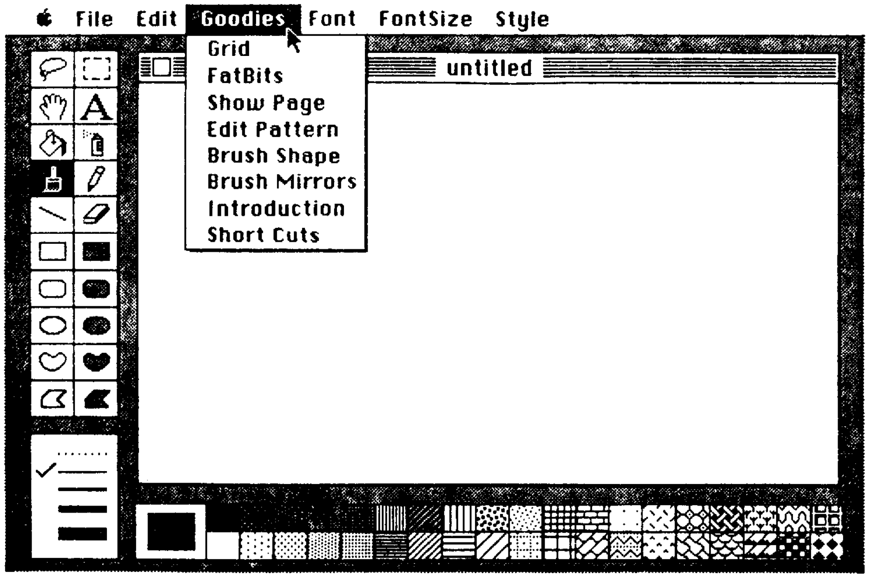

FIG. 1. Pulldown menu on the Apple Macintosh appears when the user clicks down on the mouse button.

Moving the left and right arrow keys causes items to be sequentially highlighted in reverse video. When the desired command is highlighted, pressing the RETURN key initiates the action.

Other applications of permanent menus include Apple Macpaint, computer-assisted design (CAD) systems, or other graphics systems that display an elaborate menu of commands to the side of the object being manipulated. Price [6] describes a CAD system with 120 choices in an on-screen menu. Lightpen touches or other cursor-action devices allow the user to make selections without using the keyboard.

Multiple Selection Menus A further variation on single menus is the capacity to make multiple selections from the choices offered. For example, this menu from a political interest survey allows multiple choices on one touch screen:

POLITICAL ISSUES

HIGH UNEMPLOYMENT
AID TO ELDERLY
NUCLEAR FREEZE
HIGH DEFENSE BUDGET
GOVERNMENT REGULATION
FOREIGN AID
PERSONAL TAXES
CIVIL DEFENSE
RIGHT TO ABORTION
CRIME CONTROL
MINORITY RIGHTS DONE

TOUCH UP TO THREE ISSUES THAT YOU
FEEL ARE THE MOST IMPORTANT,
AND THEN TOUCH DONE:

This situation is nicely handled with a multiple selection single menu; it would have been cumbersome to ask 11 binary choices when the user could not scan the full list of issues. The system might highlight already selected items with a check mark or bold face.

Summary Even the case of single menus provides a rich domain for designers and human factors researchers. Questions of wording, screen layout, and selection mechanism all emerge even in the simple case of choosing from one set of items. Still more challenging questions emerge from designing sequences and trees of menus.

B. Linear Sequence of Menus

Often a series of interdependent menus can be used to guide the user through a series of choices in which the user sees the same sequence of menus no matter what choices are made. A document printing package might have this linear sequence of menus:

Do you want the document printed at
  1 — your terminal
  2 — the computer center line printer
  3 — the computer center laser printer
Type the number of your choice and press RETURN:

Do you want
  1 — single spacing
  2 — double spacing
Type the number of your choice and press RETURN:

Do you want
1 — no page numbering
2 — page numbering on the top, right justified
3 — page numbering at the bottom, centered
Type the number of your choice and press RETURN:

Another example would be an online examination that had a sequence of multiple choice test items, each made up in the form of a menu.

Movement Through the Menus Linear sequences guide the user through a complex decision-making process by presenting one decision at a time. The document-printing example could be improved by offering the user a mechanism for going back to previous menus to review or change choices made earlier. A second improvement would be to display the results of previous choices, so users could see what decisions had been made. A third improvement might be to let the users know how many and which menus are yet to be seen.

The first improvement, allowing backward traversal, could be handled easily by changing the instructions to:

Type the number of your choice and press RETURN, or type "B" and press RETURN to go back to the previous menu:

The second improvement, showing the record of previous menus, could be handled by displaying the choices already made. The third improvement, showing upcoming choices, could be handled by displaying a descriptive term about the menus to follow, or simply an indication that this is the third of six menus. Unfortunately, as more improvements are made there is a greater possibility of creating cluttered displays. Judgments based on experience can resolve many decisions, but experimental tests with alternative formats and several classes of users may be useful to guide designers.

Summary Linear sequences of menus are a simple and effective means for guiding the user through a decision-making process. The user should be given a clear sense of progress or position within the sequence and the means for going backwards to earlier choices (and possibly to terminating or restarting the sequence).

Choosing the order of menus in a linear sequence is often straightforward, but care must be taken to match user expectations. One strategy is to place the easy decisions first to relieve users of some concerns, enabling them to concentrate on more difficult choices.

C. Tree-Structured Menus

When a collection of items grows and becomes difficult to maintain under intellectual control, people form categories of similar items, creating a tree structure [7,8]. Some collections can be easily partitioned into mutually exclusive groups with distinctive identifiers. Familiar examples include:

Male, female
Animal, vegetable, mineral
Spring, summer, autumn, winter
Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday
Less than 10, between 10 and 25, greater than 25
Percussion, string, woodwind, brass

Even these groupings may occasionally lead to confusion or disagreement. Classification and indexing is a complex task, and in many situations there is no perfect solution acceptable to everyone. The initial design can be improved as a function of feedback from users. Over time, as the structure is improved and as users gain familiarity with it, success rates will improve.

Despite their problems, tree-structured menu systems have the power to make large collections of data available to novice or intermittent users. If each menu has 8 items, then a menu tree with 4 levels has the capacity to lead an untrained user to the right frame out of a collection of 4096 frames.

If the groupings at each level are natural and comprehensible to the user, and if the user knows what he/she is looking for, then the menu traversal can be accomplished in a few seconds—more quickly than flipping through a book. On the other hand, if the groupings are unfamiliar and the user has only a vague notion of what he/she is looking for, it is possible to get lost in the tree menus for hours [9].

Depth Versus Breadth The depth (number of levels) of a menu tree depends, in part, on the breadth (number of items per level). If more items are put into the main menu, then the tree spreads out and has fewer levels. This is advantageous, but not if clarity is substantially compromised or if a slow display rate consumes the user's patience. Several authors have urged four to eight items per menu, but at the same time they urge no more than three to four levels. With large menu applications, one or both of these guidelines must be compromised.

D. P. Miller [10] studied user performance in retrieving items from 4 versions of a tree-structured menu system containing 64 target items. Menus had 2, 4, 8, or 64 items in each screen, with corresponding depths of 6, 3, 2, and 1. The 64 items were carefully chosen that "they form valid semantic hierarchies" in each of the 4 versions. Speed of performance was fastest with 4 or 8 items per menu, and the lowest error rate occurred with 8 items per menu. These results are useful, but there were two special conditions that may limit the applicability of this study: subjects became very familiar with the menus during the training and 128 trials, and the 64 items were chosen so that there were meaningful groupings in all 4 versions.

Kiger [11] grouped 64 items in 5 menu tree forms:

8-2: 8 items on each of 2 levels
4-3: 4 items on each of 3 levels
2-6: 2 items on each of 6 levels
4-1 + 16-1: A 4 item menu followed by a 16 item menu
16-1 + 4-1: A 16 item menu followed by a 4 item menu

Explore Litigation Insights

DOCKET ALARM

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.