



Future Vision

FUTURE VISION BIE

By K B Hemanth Raj

Visit : <https://hemanthrajhemu.github.io>

A Small Contribution Would Support Us.

Dear Viewer,

Future Vision BIE is a free service and so that any Student/Research Personal **Can Access Free of Cost.**

If you would like to say **thanks**, you can make a **small contribution** to the author of this site.

Contribute whatever you feel this is worth to you. This gives **us support** & to bring **Latest Study Material** to you. After the Contribution Fill out this Form (<https://forms.gle/tw3T3bUVpLXL8omX7>). To Receive a **Paid E-Course for Free**, from our End within 7 Working Days.

Regards

- K B Hemanth Raj (Admin)

Contribution Methods

UPI ID

1. futurevisionbie@oksbi
2. futurevisionbie@paytm

Scan & Pay

Account Transfer

Account Holder's Name: K B Hemanth Raj

Account Number: 39979402438

IFSC Code: SBIN0003982

MICR Code: 560002017

More Info: <https://hemanthrajhemu.github.io/Contribution/>



**Gain Access to All Study Materials according to VTU,
CSE – Computer Science Engineering,
ISE – Information Science Engineering,
ECE - Electronics and Communication Engineering & MORE...**

Stay Connected... get Updated... ask your queries...

Join Telegram to get Instant Updates: https://bit.ly/VTU_TELEGRAM

Contact: MAIL: futurevisionbie@gmail.com

INSTAGRAM: www.instagram.com/futurevisionbie/

WHATSAPP SHARE: <https://bit.ly/FVBIESHARE>



The Essential Guide to User Interface Design

An Introduction to GUI Design Principles and Techniques

Third Edition

Wilbert O. Galitz



Wiley Publishing, Inc.

<https://hemanthrajhemu.github.io>

Intranet Design Guidelines	258
Extranet Design Guidelines	259
Small Screens	259
Weblogs	260
Statistical Graphics	261
Types of Statistical Graphics	273
Flow Charts	283
Technological Considerations in Interface Design	284
Graphical Systems	284
Web Systems	287
The User Technology Profile Circa 2006	292
Examples of Screens	293
Example 1	293
Example 2	297
Example 3	300
Example 4	301
Example 5	302
Example 6	303
Example 7	305
Step 3 Exercise	306
Step 4 Develop System Menus and Navigation Schemes	307
Structures of Menus	308
Single Menu	308
Sequential Linear Menu	309
Simultaneous Menu	309
Hierarchical or Sequential Menu	310
Connected Menu	311
Event-Trapping Menu	313
Functions of Menus	313
Navigation to a New Menu	314
Execute an Action or Procedure	314
Displaying Information	314
Data or Parameter Input	314
Content of Menus	314
Menu Context	315
Menu Title	315
Choice Descriptions	315
Completion Instructions	315
Formatting of Menus	315
Consistency	316
Display	316
Presentation	316
Organization	317
Complexity	320
Item Arrangement	321
Ordering	321

Groupings	323
Selection Support Menus	325
Phrasing the Menu	328
Menu Titles	329
Menu Choice Descriptions	330
Menu Instructions	332
Intent Indicators	332
Keyboard Shortcuts	333
Selecting Menu Choices	337
Initial Cursor Positioning	337
Choice Selection	338
Defaults	339
Unavailable Choices	340
Mark Toggles or Settings	340
Toggled Menu Items	341
Web Site Navigation	342
Web Site Navigation Problems	343
Web Site Navigation Goals	344
Web Site Navigation Design	345
Maintaining a Sense of Place	367
Kinds of Graphical Menus	369
Menu Bar	369
Pull-Down Menu	371
Cascading Menus	375
Pop-Up Menus	377
Tear-Off Menus	379
Iconic Menus	380
Pie Menus	380
Graphical Menu Examples	382
Example 1	382
Step 5	
Select the Proper Kinds of Windows	385
Window Characteristics	385
The Attraction of Windows	386
Constraints in Window System Design	388
Components of a Window	390
Frame	390
Title Bar	391
Title Bar Icon	391
Window Sizing Buttons	392
What's This? Button	393
Menu Bar	393
Status Bar	394
Scroll Bars	394
Split Box	394
Toolbar	394
Command Area	395

Develop System Menus and Navigation Schemes

A system contains large amounts of information and performs a variety of functions. Regardless of its purpose, the system must provide some means to tell people about the information it possesses or the things it can do. This is accomplished by displaying listings of the choices or alternatives the user has at appropriate points while using the system; or creating a string of listings that lead a person from a series of general descriptors through increasingly specific categories on following listings until the lowest level listing is reached. This lowest level listing provides the desired choices. These listings of choices are commonly called menus. Menus are a major form of navigation through a system and, if properly designed, assist the user in developing a mental model of the system. In this step, the following menu topics will be addressed:

- The structures of menus.
- The functions of menus.
- The content of menus.
- Formatting menus.
- Writing menus.
- Navigation using menus.
- Web site navigation and links.
- Web site navigation elements.
- Maintaining a sense of place in Web sites.
- Types of menus.

Menus are effective because they utilize the more powerful human capability of recognition rather than the weaker capability of recall. Working with menus reminds people of available options and information that they may not be aware of or have forgotten.

Menus are not without problems, however. New and inexperienced system users might find learning large systems difficult. Menu information must often be remembered and integrated across a series of screens. If each menu is viewed in isolation, relationships between menus may be difficult to grasp. Words and phrases with multiple meanings may also be wrongly interpreted because of the user's inability to see relationships between menus. Ambiguities, also, may not be correctly resolved if the user makes incorrect assumptions about menu structure. The frequent result is that people make mistakes and can get lost in the hierarchical structure.

Experienced system users, while finding menus helpful at first, may find them tedious as they learn the system. Continually having to step through a series of menus to achieve the desired objective can be time-consuming and frustrating. Therefore, the design of menu systems must consider the conflicting needs of both inexperienced and experienced users.

Graphical and Web systems are heavily menu-oriented. They vary in form and are applied in diverse ways. In graphical systems they are used to designate commands, properties that apply to an object, documents, and windows. When selected, a graphical menu item may lead to another menu, cause a window to be displayed, or directly cause an action to be performed. To accomplish these goals, a graphical system presents a variety of menu styles to choose from. Included are entities commonly called menu bars, and menus called pull-downs, pop-ups, cascades, tear-offs, and iconic. In Web site design, common menus include textual links to other pages, command buttons, and both graphical and textual toolbars.

In this step, graphical and Web system menus will be addressed. It will begin with a description of the kinds of menu structures available and their content, and then present a series of general menu design guidelines for formatting, phrasing, selecting choices, and navigating menus. Next, Web-specific navigation issues and guidelines will be discussed. Finally, specific types of graphical menus will be described, recommendations for proper usage given, and relevant specific guidelines summarized.

Structures of Menus

Menus vary in form from very simple to very complex. They may range from small dialog boxes requesting the user to choose between one of two alternatives, to hierarchical tree schemes with many branches and levels of depth. A menu's structure defines the amount of control given to the user in performing a task. The most common structures are the following.

Single Menu

In this simplest form of menu, a single screen or window is presented to seek the user's input or request an action to be performed, as illustrated in Figure 4.1. In using

the Internet, for example, at a point in the dialog people may be asked if they wish to “Stay Connected” or “Disconnect.” In playing a game, choices presented may be “novice,” “intermediate,” or “expert.” Single menus conceptually require choices from this single menu only, and no other menus will follow necessitating additional user choices. The user need only consider the immediate consequences of the item being chosen and need not be concerned with any other additional system menus. While other single menus may exist in the system and might be encountered later, these other menus are not perceived by the user as comprising a series of choices.

A single menu may be iterative if it requires data to be entered into it and this data input is subject to a validity check that fails. The menu will then be represented to the user with a message requesting reentry of valid data.

Sequential Linear Menus

Sequential linear menus are presented on a series of screens possessing only one path. The menu screens are presented in a preset order, and, generally, their objective is for specifying parameters or for entering data. The length of the path may be short or long, depending upon the nature of the information being collected. All the menus are important to the process at hand and must be answered in some manner by the user. Sequential linear menus are illustrated in Figure 4.2.

Sequential path menus have several shortcomings. A long sequence may become tedious as menu after menu is presented. The user may not remember an answer to a previous question, a question important to the currently presented choices. The user may also want to return to a previous menu to change an answer or look at an answer, an awkward process that must be allowed. Finally, the user may, conceptually, want to complete the menus in a different order than that in which they are being presented.

Simultaneous Menus

Instead of being presented on separate screens, all menu options are available simultaneously, as illustrated in Figure 4.3. The menu may be completed in the order desired by the user, choices being skipped and returned to later. All alternatives are visible for reminding of choices, comparing choices, and changing answers. The tedium associated with a long series of sequential menus is greatly reduced.

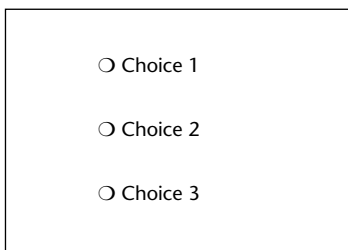


Figure 4.1: Single menu.

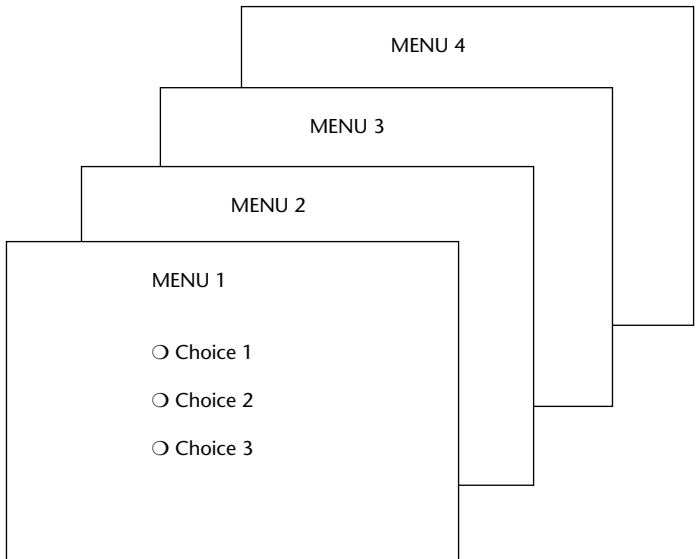


Figure 4.2: Sequential linear menus.

ALTERNATIVE 1	ALTERNATIVE 3
<ul style="list-style-type: none">○ Choice 1○ Choice 2○ Choice 3	<ul style="list-style-type: none">○ Choice 1○ Choice 2○ Choice 3
ALTERNATIVE 2	ALTERNATIVE 4
<ul style="list-style-type: none">○ Choice 1○ Choice 2○ Choice 3○ Choice 3	<ul style="list-style-type: none">○ Choice 1○ Choice 2○ Choice 3○ Choice 3

Figure 4.3: Simultaneous menus.

Problems with simultaneous menus are that for large collections of menu alternatives screen clutter can easily occur, and screen paging or scrolling may still be necessary to view all the choices. This type of menu must also clearly indicate menu choice relationships and dependencies, something better accomplished in a linear menu string, or a hierarchical menu, described next. Presenting many menu dependencies and relationships on a screen, especially if poorly indicated, can also be very confusing for a novice user.

Hierarchical or Sequential Menus

When many relationships exist between menu alternatives, and some menu options are only appropriate depending upon a previous menu selection, a hierarchical structure is the best solution. In Web site design, hierarchical menus are often referred to as

sequential menus. A hierarchical structure results in an increasing refinement of choice as menus are stepped through, for example, from options, to suboptions, from categories to subcategories, from pages to sections to subsections, and so on. A hierarchical structure can best be represented as an inverse tree, leading to more and more branches as one moves downward through it. Hierarchical structures are characterized by depth and breadth, depth being the number of choice levels one must traverse to reach the destination, breadth being the number of alternatives found at each level. Menu depth and breadth has been a well-researched topic and will be fully discussed in succeeding pages. Common examples of hierarchical design today are found in menu bars with their associated pull-downs, and in Web sites with their navigation links.

The order and structure of branching in a hierarchy is preset and the normal order of flow one-way: top down. A disadvantage of a hierarchical scheme is that the defined branching order may not fit the user's conception of the task flow. If users are not familiar with the hierarchical menu, or are unable to predict what suboptions lie below a particular choice, they may go down wrong paths and find it necessary to go back up the tree to change a choice, or perhaps even return to the top-level menu. It is important, then, that hierarchies be consistent with user expectations, and that choice uncertainties be reduced as much as possible. It must also be easy to back upward through the tree to facilitate exploration of the tree.

Hierarchical menus are illustrated in Figure 4.4. Note that the top level of the tree is considered level 0 with subsequent levels numbered sequentially beginning with number 1. Starting at the top, level 0, two selections, or mouse clicks, are required to reach level 2.

Connected Menus

Connected menus are networks of menus all interconnected in some manner. Movement through a structure of menus is not restricted to a hierarchical tree, but is permitted between most or all menus in the network. From the user's perspective there is no top-down traversal of the menu system, but an almost unhindered wandering between any two menus of interest. A connected menu system may be cyclical, with movement permitted in either direction between menus, or acyclical, with movement permitted in only one direction. These menus also vary in connectivity, the extent to which menus are linked by multiple paths. (In a hierarchical menu system, the ability to go back to a previous menu or to return to the top-level menu are also examples, although restricted, of connected menus.)

The biggest advantage of a connected menu network is that it gives the user full control over the navigation flow. Its disadvantage is its complexity, and its navigation may be daunting for an inexperienced user. An example connected menu structure is represented in Figure 4.5.

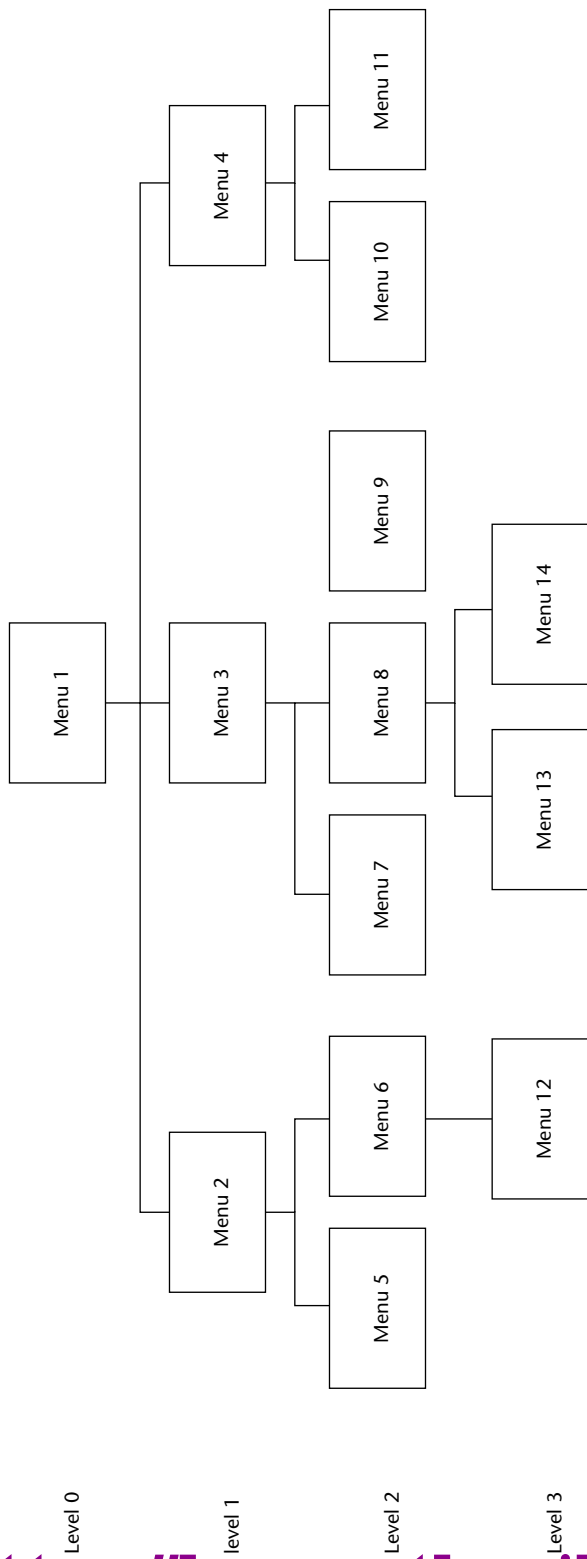


Figure 4.4: Hierarchical or sequential menus.

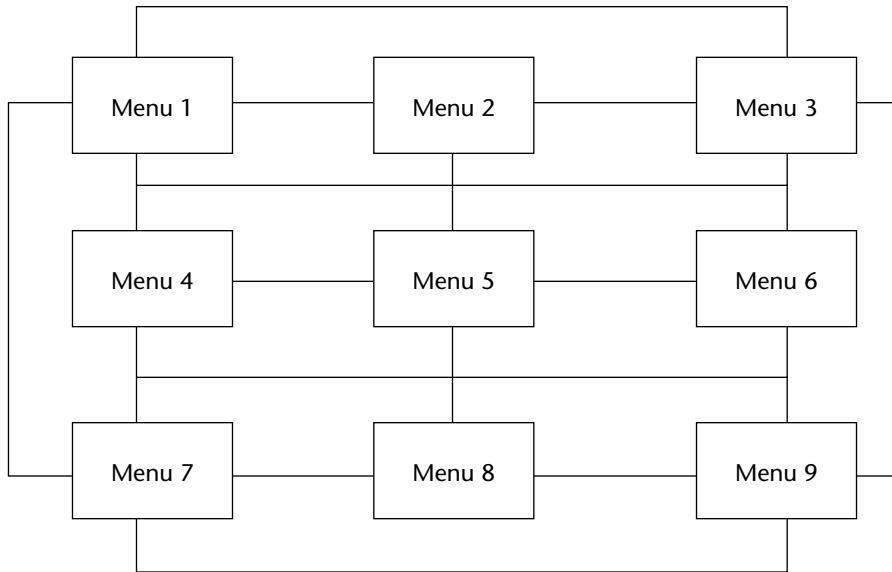


Figure 4.5: Connected menus.

Event-Trapping Menus

Event-trapping menus provide an ever-present background of control over the system's state and parameters while the user is working on a foreground task. They are, in essence, a set of simultaneous menus imposed on hierarchical menus. In a graphical system, for example, existing together are a simultaneous menu, the menu bar, and a hierarchy — the menu bar and its pull-downs. Event-trapping menus generally serve one of three functions; (1) They may immediately change some parameter in the current environment (bold a piece of text), (2) they may take the user out of the current environment to perform a function without leaving the current environment (perform a spell check), or (3) they may exit the current environment and allow the user to move to a totally new environment (Exit).

These menus can also change content based upon the system state, or an event, existing at that moment. A Paste option in a word-processing application, for example, will only function if there is something in a clipboard to paste. A Grid option on a pull-down, as another example, will toggle between a "Hide Grid" or "Show Grid" state, depending upon whether or not a grid is displayed on the screen at that moment. Event-trapping menus such as menu bars are constantly available to aid in establishing a sense of context, or where one is, while things may be changing in the foreground.

Functions of Menus

From the user's perspective, a menu can be used to perform several functions: to navigate to a new menu, to execute an action or procedure, to display information, or to input data or parameters.

Navigation to a New Menu

Each user selection causes another menu in a hierarchical menu tree to be displayed. The purpose of each selection is to steer the user toward an objective or goal. Selection errors may lead the user down wrong paths, and cost time and, perhaps, aggravation, but these errors are nondestructive and usually undoable.

Execute an Action or Procedure

A user selection directs the computer to implement an action or perform a procedure. The action may be something like opening or closing a file, copying text, or sending a message. In some cases, execution may only occur after a hierarchical menu tree is navigated. In other cases, actions may be performed as successive hierarchical menus are encountered and traversed. Selection errors may or may not have serious consequences, depending upon the nature of the action. Accidental selection of critical irreversible actions must be prevented in interface design.

Displaying Information

The main purpose of selecting a menu choice may simply be to display information. The user may be searching for specific information in a database or browsing the Web. The user's focus is primarily on the information desired and less on the selection function. In many cases, information retrieval may occur only after a hierarchical menu tree is navigated. The content material and the user's interests will determine the paths followed. Users may spend considerable time and effort understanding and processing uncovered information to evaluate subsequently displayed menu choices. Wrong turns in the process will again cost time and perhaps aggravation, but these errors are nondestructive and usually undoable.

Data or Parameter Input

Each selection specifies a piece of input data for the system or provides a parameter value. Data or values may be input on a single menu or spread over a hierarchy of menus. The user's focus is primarily on the information being provided and, again, less on the selection function. Selection errors can be easily corrected if detected by the system.

Content of Menus

A menu consists of four elements: its *context*, its *title*, its *choice descriptions*, and its *completion instructions*. These concepts are introduced here and will be expanded in detailed guidelines to follow on succeeding pages.

Menu Context

A menu's context provides information to keep the user oriented. This kind of information is critical in complex or hierarchical menu systems, where loss of position or disorientation can easily occur. Feedback is necessary that tells users where they are in a process, what their past choices were, and possibly how much farther they still have to navigate. Human memory being what it is, where one is and how one got there all too easily slip from consciousness.

Verbal linkage, spatial linkage, or both may be used to provide navigation feedback. *Verbal linkage* involves providing, on the current menu screen, a listing of choices made on previous menus that have led to this position. It also involves assuring the user that the displayed menu is the menu desired. Its title should mirror the option selected on the previous menu, and its content should reflect its title. *Spatial linkage* can be accomplished by graphic methods. Each succeeding menu screen can be displayed overlapping the previous menu screen so a succession of choices can be seen in a single view. A sense of progress and distance can then be easily ascertained.

Menu Title

A menu's title provides the context for the current set of choices. The title must reflect the choice selected on the previously displayed menu.

Choice Descriptions

Choice descriptions are the alternatives available to the user. These descriptions can range from a mnemonic, numeric, or alphabetized listing of choices, to single words or phrases, to full sentences, or more. The style chosen will reflect the experience of the user (novice or expert), the nature of the choices (well-learned alternatives or not), the nature of the selection mechanism (keyboard or mouse), and the nature of the system (business system application or Web page).

Completion Instructions

Completion instructions tell users how to indicate their choices. They may include the rationale for why the user is being asked to make this choice and the impact the choice will have on subsequent processes. Explicit instructions may be needed for first time or casual users of a system. Experienced users will find overly verbose instructions unnecessary. The needs of all system users, and the nature of the system, must again be considered in creating this kind of on-screen guidance.

Formatting of Menus

The human-computer interface has a rich history of experimental studies with menus, the results of which can and have been applied to graphical screen and Web page menu design and presentation. What follows is a series of guidelines for formatting menus.

Consistency

- Provide consistency with the user's expectations.
 - Provide consistency in menu
 - Formatting, including organization, presentation, and choice ordering.
 - Phrasing, including titles, choice descriptions, and instructions.
 - Choice selection methods.
 - Navigation schemes.
-

Like all aspects of screen design, menu design consistency is an integral component of system usability. Menu formatting, phrasing, choice selection, and navigation must be consistent throughout a graphical system or Web site.

Display

- If continual or frequent references to menu options are necessary, permanently display the menu in an area of the screen that will not obscure other screen data.
 - If only occasional references to menu options are necessary, the menu may be presented on demand.
 - Critical options should be continuously displayed, however.
-

Whether to display a menu continually, or on demand, is determined by the menu's frequency of use. Always permanently display menus that are frequently referenced. This will provide memory support and immediate access to what is needed most. Occasionally needed menus may be presented on request via pop-ups or pull-downs. Critical options, however, should always be continuously displayed. Wright, Lickorish, and Milroy (1994) found superior performance for permanently displayed menus as opposed to menus that had to be retrieved through mouse clicks. They speculate that because retrieving a menu for display requires more actions, this may also impair people's memory for other tasks being performed.

Presentation

- Ensure that a menu and its choices are obvious to the user by presenting them with a unique and consistent structure, location, and/or display technique.
 - Ensure that other system components do not possess the same visual qualities as menu choices.
-

A menu and its choices should be immediately recognizable by the user as being a menu of choices. This can be accomplished through giving the menu a distinctive and consistent structure and presenting it in a consistent screen or page location. Presentation techniques must, of course, be compatible with those used for other purposes on the remainder of the screen. A good way to set a menu off from the remainder

of the screen is to enclose it in a box or display it using a background that contrasts with the remainder of the screen. Techniques chosen should be consistent throughout the system. Web page navigation links, which may be scattered throughout a page, are displayed underlined and in a unique color to differentiate and identify them.

Ensure that other system elements do not possess qualities that allow users to confuse them with menu choices. In Web page design, for example, the underlining of any system component other than navigation links is not recommended because of the possibility that they may be confused with links.

Organization

- Provide a general or main menu.
 - Display
 - All relevant alternatives.
 - Only relevant alternatives.
 - Delete or gray-out inactive choices.
 - Match the menu structure to the structure of the task.
 - Organization should reflect the most efficient sequence of steps to accomplish a person's most frequent or most likely goals.
 - Minimize number of menu levels within limits of clarity.
 - For Web sites, restrict it to two levels (requiring two mouse clicks) for fastest performance.
 - Be conservative in the number of menu choices presented on a screen.
 - Without logical groupings of elements, limit choices to 4 to 8.
 - With logical groupings of elements, limit choices to 18 to 24.
 - Provide decreasing direction menus, if sensible.
 - Never require menus to be scrolled.
-

In organizing a menu, the goal is to simply and effectively reveal its structure, while also reducing the number of actions needed to locate the target item.

General menu. The top-level menu in a hierarchical menu scheme should be a general or main menu, consisting of basic system options. This will provide a consistent starting point for all system activities and a “home base” to which the user may always return.

Relevant alternatives. A menu should provide all relevant alternatives, and only relevant alternatives, at the point at which it is displayed. Including irrelevant choices on a menu screen increases learning requirements and has been found to interfere with performance. There are two exceptions to this rule, however. Alternatives that are conditionally inactive may be displayed along with the conditionally active choices, if the active choices can be visually highlighted in some manner (such as through bolding or reverse video), or the inactive choices can be visually subdued (perhaps as through graying them out). One study, however, found that completely eliminating inactive alternatives on a menu resulted in faster choice access time, when compared to leaving inactive alternatives on a

menu, but displayed in a subdued manner. This study concludes that eliminating conditionally inactive choices from a menu appears to be the best approach. Mayhew (1992) suggests that while deletion does provide an advantage to expert users of keyboard-driven menus, graying out seems to be advantageous to novices in systems using pointer-driven selection devices. She concludes that because menus are geared toward novices, graying appears to be the best overall choice. In general today's graphical systems follow the gray-out approach for inactive menu choices. Whatever method is chosen should be consistently followed throughout a system. Options to be implemented in the future may also be displayed if they can be visually marked in some way (through a display technique or some other annotation).

Matching menu structure to the tasks. Menus should be organized according to how people structure their tasks. They should reflect the most efficient sequence of steps to accomplish a person's most frequent or likely goals.

Minimize number of levels. The issue that must be addressed in creating a multi-level menu structure is determining how many items will be placed on one menu (its breadth) and how many levels it will consume (its depth). In general, the more choices contained on a menu (greater breadth), the less will be its depth; the fewer choices on a menu (less breadth), the greater will be its depth.

The advantages of a menu system with greater *breadth* and less depth are

- Fewer steps and shorter time to reach one's objective.
- Fewer opportunities to wander down wrong paths.
- Easier learning by allowing the user to see relationships of menu items.

A broad menu's disadvantages are

- A more crowded menu that may reduce the clarity of the wording of choices.
- Increased likelihood of confusing similar choices because they are seen together.

The advantages of greater *depth* are

- Less crowding on the menu.
- Fewer choices to be scanned.
- Easier hiding of inappropriate choices.
- Less likelihood of confusing similar choices because there is less likelihood that they will be seen together.

Greater depth disadvantages are

- More steps and longer time to reach one's objective.
- More difficulties in learning because relationships between elements cannot always be seen.
- More difficulties in predicting what lies below, resulting in increased likelihood of going down wrong paths or getting lost.
- Higher error rates.

In text-based and graphical systems, a good number of studies have looked at the breadth-depth issue. Included are studies by Kiger (1984) and Jacko and

Salvendy (1996). Some have concluded that breadth is preferable to depth in terms of either greater speed or fewer errors, that a low number of levels (two to three) and an intermediate number of choices (four to eight) results in faster, more accurate performance as opposed to fewer or greater numbers of levels and choices, and that four to eight choices per menu screen is best. Another study found that one level was easiest to learn, and a couple of studies have concluded that a menu could contain up to 64 items if it were organized into logical groups. The least desirable alternative in almost all cases was deep-level menus that simply presented the user with a binary choice (select one of two alternatives) on each menu. In an early study of hypertext, Snowberry et al. (1983) found that as hypertext depth increased, performance and preference declined and errors increased. The general conclusion of these studies:

- People found resources faster in, and understood better, broader, shallow menu structures than narrow deep ones.

In Web site design, studies have also looked at the breadth-depth issue (Zaphiris and Mtei, 1998; Larson and Czerwinski, 1998). Both found that a two-level (two mouse click) Web site was searched faster than those containing more levels. Zaphiris (2001) modeled user performance in menu search using the results of previous studies and concluded that a menu design of either extreme (very deep or very broad) undermines learnability and usability.

Straub (2003a) describes additional research that suggests a concave hierarchical menu structure is optimal for browsing (Norman and Chin, 1988; Bernard, 2002). A concave shape presents a broad initial selection screen followed by menus with small categories. The terminal menu then contains an option set that is again somewhat broad. Straub's broad conclusion from the research is this:

- **Too deep is too deep.** People have a more difficult time understanding, and consequently navigating, deep sites.
- **Too broad is too broad.** Extremely broad sites also present a challenge to efficient navigation.
- **Effective sub-grouping reduces perceived breadth.** Logical grouping improves performance for even the broadest structures.
- **Clear labels improve navigation accuracy.** Creating clear and distinct labels for navigation elements enhances performance (as described in Step 3.)

The conclusion that one might derive from these studies is this: Fewer levels of menus aid the decision-making process, but trying to put too many choices on a single menu also has a negative impact. The final solution is a compromise: Minimize the number of levels within limits of clarity. What is clarity? The studies seem to indicate that, if the choices to be displayed cannot be segmented into logical categories, then confine the number of alternatives displayed to four to eight per menu. (Straub suggests a maximum of 16 for links.) If logical categorization is possible, and meaningful, logical category names can be established, then a larger number of choices can be presented. The maximum number of alternatives will, however, be dependent upon the size of the words needed to describe the alternatives to the user. Wordy captions will greatly restrict the number of alternatives capable of being displayed. There is one exception to these

basic principles. Large, linearly ordered, well-learned listings, such as months of the year, or numbers, would be better presented in a one-level menu, rather than by breaking them into multiple levels.

Limit the number of choices. Be conservative in the number of menu choices presented on a screen. If the choices cannot be logically grouped, restrict the number to four to eight as just described. If the choices can be grouped, 18 to 24 can be displayed, with no more than 10 items within a group. Mayhew (1992) suggests that if the menu choices are complex and/or there are no groupings of items, choices presented should be restricted to *10 or fewer*. This recommendation is similar to the eight or fewer recommendations above. If the menu choices are not complex, items on the menu can be grouped, and the users are infrequent or casual, she recommends *20 or fewer* choices. If the menu choices are not complex, items on the menu can be grouped, and the users are frequent or expert, she suggests *21 or more* choices can be provided.

MAXIM The best journey is the one with the fewest steps.

Provide decreasing direction menus. In addition to breadth and depth, direction has been found to affect menu choice selection performance. In a multilevel menu, a *decreasing* direction structure presents successively fewer choices as each lower level is traversed. An *increasing* direction structure presents successively more choices as each lower level is traversed. Bishu and Zhan (1992), in a study of 16 and 32 item iconic menus, found that decreasing direction menus were significantly faster and more accurate than increasing menus. As just described, however, the research of Norman and Chin (1988) and Bernard (2002) suggests a concave hierarchical menu structure is optimal for browsing.

Scrolling. Never require menus to be scrolled. Keep all choices visible at all times.

Complexity

- Provide both simple and complex menus.
 - Simple: a minimal set of actions and menus.
 - Complex: a complete set of actions and menus.
-

Providing two sets of menus will more effectively satisfy the differing needs of the novice and expert user. The novice or casual user often only requires a minimal set of actions and menus to accomplish tasks. The expert may prefer a full set of options. Make selection, and changing, between simple and complex menus easy to accomplish, preferably through a menu bar choice. IBM's SAA CUA refers to these menus as *short* and *full*.

Item Arrangement

- Align alternatives or choices into single columns whenever possible.
 - Orient for top-to-bottom reading.
 - Left-justify descriptions.
 - If a horizontal orientation of descriptions must be maintained
 - Organize for left-to-right reading.
-

For scanning ease, menu choices should be left-justified and aligned vertically into columns. Research has found that columnar menus and listings are searched much faster than horizontally-oriented menus. Do not array a menu in multiple columns.

When menus are included on other screens, space constraints often exist, and the menu must be arrayed horizontally. If a single-row (horizontal) orientation is necessary organize for left-to-right reading. If two or more rows are available for displaying choices, organize for top-to-bottom, then left-to-right reading to facilitate visual scanning.

Ordering

- Order lists of choices by their natural order, or
 - For lists associated with numbers, use numeric order.
 - For textual lists with a small number of options (seven or less), order by
 - Sequence of occurrence.
 - Frequency of occurrence.
 - Importance.
 - Semantic similarity.
 - Use alphabetic order for
 - Long lists (eight or more options).
 - Short lists with no obvious pattern or frequency.
 - Separate potentially destructive actions from frequently chosen items.
 - Maintain a consistent ordering of options on all related menus.
 - For variable-length menus, maintain consistent relative positions.
 - For fixed-length menus, maintain consistent absolute positions.
-

Within information categories included on a menu, or in menus in which categories are not possible, options must be ordered in meaningful ways. When a menu contains multiple categories of information, the ordering of categories will follow these same principles. A meaningful ordering is necessary to

- Facilitate search for an item.
- Provide information about the structure and relationships among items.
- Provide compatibility with the user's mental model of the item structure.
- Enhance the user's ability to anticipate a choice's location.

When items are organized along some dimension or characteristic, the user can use that information to locate items faster. An alphabetized list, for example, provides an indication of approximately where in the listing an item beginning with a particular letter will be found. Understanding structure and relationships, item similarities and dissimilarities, can also aid in focusing attention on that which is relevant. Any incompatibility with the user's mental model will disrupt searching as the user tries to make sense of something that had been well understood, but now is being presented in a way that has not been well learned. Months of the year presented in alphabetic order, for example, would be very disrupting.

Experienced users often anticipate the location of a desired choice within a familiar menu. Hornof and Kieras (1999), in studying how items are selected from pull-down menus, found that people often make an initial eye and mouse-positioning movement toward the expected choice location before the pull-down even appeared on the screen. They also found that choices in the top three positions of the pull-down were selected faster than those in other positions. This may have been caused by users' ability to better predict a choice's location at the top, and/or the shorter mouse movement required from the menu bar to the pull-down. Observational studies also reveal that experienced users also anticipate the location of command buttons appearing within a window. While waiting for a window to appear upon which a command button will be immediately "clicked," the pointer is often positioned at the button's expected location before the window appears.

Another study, Byrne, John, Wehrle, and Crow (1999), studied how people search unfamiliar pull-down menus. They found that the search primarily flowed from menu top to bottom, and that the initial eye fixation was usually focused on the choice in the topmost menu position. Almost all recorded eye fixations were on one of the first three items.

Both of the studies point out the importance of presenting important menu items at the top of menu arrays, and providing consistency in menu organization schemes and menu locations. Common ordering schemes for menus, then, are the following:

Natural ordering. If items have a natural sequence, such as chapters in a book, days in a week, or months in the year, the ordering scheme should follow this natural sequence. The screen viewer will have learned these ordering schemes very well.

Numeric ordering. Use numeric ordering for choices associated with numbers, for example, type size, baud rate, or number of pixels.

Small number of options. For groupings with a small number of options (about seven or fewer), *sequence of use*, *frequency of use*, or *importance* are good ordering schemes. Also consider ordering by *semantic similarity*, along a semantic dimension such as impact, potency, or emphasis. Type style, for example, may be ordered by emphasis from least to most: regular, underlined, italicized, and bold.

Alphabetic order. For a large number of options, alphabetic ordering of alternatives is desirable. Alphabetic ordering is also recommended for small lists where no frequency or sequence pattern is obvious. It has been found that alphabetically ordered menus can be searched much faster than randomly ordered menus. One study, for example, found that an 18-item alphabetic menu was visually searched four times faster than a randomly organized menu. Search time was a function of

saccadic eye movements through the display. Search patterns were random, but fewer eye movements were required with the alphabetic arrangement. After 20 trials, however, only one eye movement was required for all conditions, and search time was the same. Another study has found that the longer the list, the greater the value of an alphabetic ordering scheme. As list length increased, the time to find items in longer random lists increased significantly faster than the time to find items in longer alphabetic lists. Learning of a randomly ordered menu will eventually take place, but this learning will be greatly aided by a meaningful choice-ordering scheme.

Separate destructive choices. Destructive menu choices, such as delete or clear, should be positioned as far away from frequently chosen choices as possible to minimize the chance of accidental selection.

Consistency between menus. Options found on more than one menu should be consistently positioned on all menus. If menus are of variable length, maintain relative positioning of all item options (for example, always place Exit at the bottom or end of the list). If menus are of fixed length, place options in the same physical position within the list.

Groupings

-
- Create groupings of items that are logical, distinctive, meaningful, and mutually exclusive.
 - Categorize them in such a way as to
 - Maximize the similarity of items within a category.
 - Minimize the similarity of items across categories.
 - Present no more than six or seven groupings on a screen.
 - Order categorized groupings in a meaningful way.
 - If meaningful categories cannot be developed and more than eight options must be displayed on a screen, create arbitrary visual groupings that
 - Consist of about four or five, but never more than seven, options.
 - Are of equal size.
 - Separate groupings created through either
 - Wider spacing, or
 - A thin ruled line.
 - Provide immediate access to critical or frequently chosen items.
-

Create groupings. Items displayed on menus should be logically grouped to aid learning and speed up the visual search process. Studies have demonstrated that logically categorized menus are easier to learn and result in faster and more accurate performance. Categorical organization may facilitate the transition from novice to expert user because information is visually represented in the way people think about it.

Categorizing. Groupings should also cover all the possibilities and contain items that are non-overlapping. While some collections of information will be easily partitioned into logical groups, others may be very difficult to partition. Some users may not understand the designer's organizational framework, and there may be differences among users based on experience. Thus, no perfect solution may exist for all, and extensive testing and refinement may be necessary to create the most natural and comprehensible solution.

Number. Limit the number of groupings on a screen to six or seven. The total number of items within all the groupings should not exceed about 18 to 24.

Ordering. Groupings of menu items may be ordered following the guidelines described in "Ordering" earlier in this step. Ordering alternatives include alphabetic, sequence of use, frequency of use, importance, and semantic similarity.

Arbitrary visual groupings. Uncategorized menus should be broken in arbitrary visual groupings through the use of space or lines. Groups should be as equal in size as possible and consist of about four or five options. Groupings should never exceed more than seven options.

Separation. Perceptually separate groupings by leaving a wider spacing between groupings, or by inscribing line separators between groupings. Guidelines for displaying line separators follow.

Critical choices. Choices that are critical or frequently chosen should be accessible as quickly and through as few steps as possible. Place them on the highest-level menu, whenever possible.

Line Separators

- Separate vertically arrayed groupings with subtle solid lines.
 - Separate vertically arrayed subgroupings with subtle dotted or dashed lines.
 - For subgroupings within a category,
 - Left-justify the lines under the first letter of the columnized choice descriptions.
 - Right-justify the lines under the last character of the longest choice description.
 - For independent groupings,
 - Extend the line to the left and right menu borders.
-

Inscribing subtle solid or dashed lines between groupings can reinforce groupings and subgroupings of vertically arrayed related choices. For breaking subgroupings within one category, the line or lines should only extend from the first character of the descriptions to the end of the longest description, as shown in Figure 4.6. Many graphical platforms always extend the line from menu border to border, as illustrated in Figure 4.7. This extended line results in too strong a visual separation between what are related menu parts. Visual separation should exist, but it should not be too overpowering.

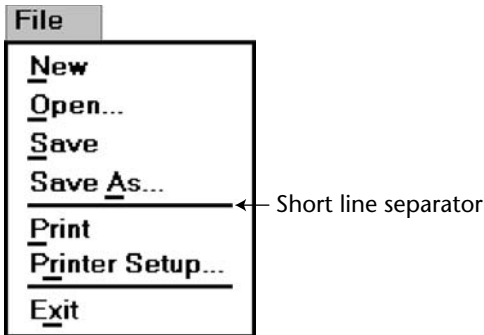


Figure 4.6: Partial line separators.

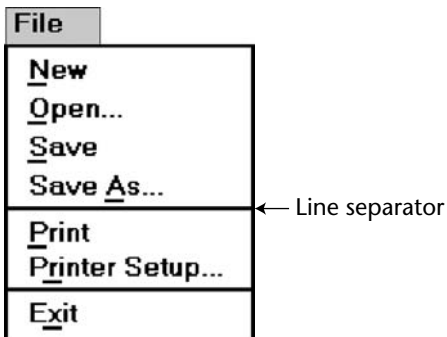


Figure 4.7: Extended line separators.

For independent groups of choices, extend the horizontal line from menu border to border. This will indicate to the user that the groupings are independent of one another. In summary, use a partial line for separating related choices; use an extended line for separating unrelated or independent choices.

Selection Support Menus

- When a small, discrete set of functions is accessed 90 percent or more of the time
— Use Folded menus.
- When a small set of items is selected between 31 percent and 89 percent of the time and the other items are selected with lower frequencies
— Use Split menus.
- If there is no small, discrete set of items that is used 30 percent of the time or more
— Use traditional menus.
- Do not reorder menus.

Selection support menus support the user by providing faster access to frequently used functions. More frequent choices are positioned at the top of a menu where a person's eye initially goes when a menu is presented. High-frequency items at the top may be specified by the menu designer and remain static or unchangeable, or the menu may be restructured reflecting a person's use pattern over a period of time. Changeable menus are called *adaptive* menus. Duplicating the more frequent items in a separate section at the top is commonly called a *split* menu. Sears and Shneiderman (1994) showed that split menus were better than traditional menus both performance-wise and preference-wise. A traditional menu is illustrated in Figure 4.8, a split menu in Figure 4.9.

A version of the split menu is one called the *folded* menu. In this menu the high-frequency items appear first and alone. The complete menu appears after a time delay, or after the user clicks on a down arrow at the bottom of the initial menu. The additional choices are displayed "below the fold" as illustrated in Figure 4.10.

In addition to providing faster access to choices in a location where the user first looks, it is argued that adaptive menus aid user learning by simplifying menus. Also, where the menu changes are made based upon user behavior, a user's exact needs are anticipated and more completely fulfilled.

These menus may also create problems for users. If lower frequency items are incorrectly chosen for placement above the split or fold, high-frequency items placed below will take longer to perform. When items are relocated to different positions, a person's spatial memory is impacted. If users do not see the additional menu items, system learning may also be degraded. Some studies do show that users dislike the extra click or delay imposed by folded menus (Card, 1982; Somberg, 1987).

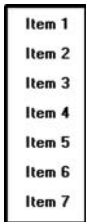


Figure 4.8: Traditional menu.

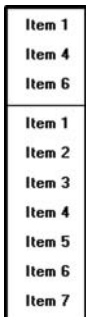


Figure 4.9: Split menu.

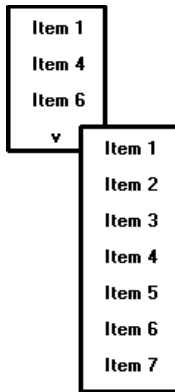


Figure 4.10: Folded menu.

To address some of the issues, Lee and Yoon (2004) conducted a study to determine when it was best to use various kinds of these menus. They evaluated traditional (static) menus, split menus, folded menus, and a fourth type they called *temporal*. A temporal menu is a traditional menu that first presents only the high-frequency items in their regular menu positions. After a short delay, the remaining lower-frequency items are filled in their normal positions as illustrated in Figure 4.11.

The study task was to select randomly selected items from menus containing seven choices. The menus were not adaptive in the sense that they were not reordered to reflect the choices being made. Two-thirds of the way through the test, however, the items were reordered to see the impact. The conclusion:

- Split menus had the fastest overall performance and were liked the best.
- For high-frequency items, split and folded menus were about equal.
- Performance on folded menus declined fastest as selection frequency went down (a wide range of functions being regularly used).
- After the item order switch, performance for both split and folded menus was poorer than that for the traditional and temporal menus.

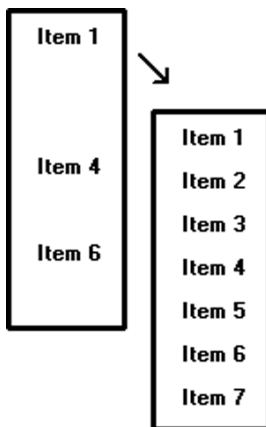


Figure 4.11: Temporal menu.

Based upon the results of this research Lee and Yoon developed a network to model the impact of selection frequency on selection times. The results, as presented in the preceding guidelines, are as follows:

- When a small, discrete set of functions is accessed 90 percent or more of the time, use folded menus.
- When a small set of items is selected between 31 percent and 89 percent of the time and the other items are selected with lower frequencies, use split menus.
- If there is no small, discrete set of items that is used 30 percent of the time or more, use traditional menus.

Although adaptivity is thought to be a desirable quality of a computer system, this appears to not be so for menu option ordering. Both split and folded menus in this study resulted in poorer performance after item reorganization. Another study compared static or fixed menus with dynamic menus whose options were continually reordered based upon the frequency in which they were chosen. Dynamic menus were slower to use and less preferred than static menus. The continual reordering interfered with menu order learning.

Phrasing the Menu

A menu must communicate to the user information about

- The nature and purpose of the menu itself.
- The nature and purpose of each presented choice.
- How the proper choice or choices may be selected.

Writing the content of menu components, the menu's title, the choice descriptions, and instructions, is often made difficult because of the varying experience levels of the menu users. At one extreme, there is the desire to explain, on the screen, everything in great detail. On the other hand, brevity is also important because of screen space constraints and limits on what people want to read. These conflicting goals often cause a trade-off between thoroughness and brevity. Also important in hierarchical menu systems is the role menus play in enabling a person to maintain a sense of place, or "Where am I now?" Also very important is a menu's ability to enable the user to accurately predict where a choice will lead, or what it will cause to happen, preventing user tedium and frustration. So, the menu content must be informative, but not intrusive. And it must balance the needs of all its expected users.

Following are guidelines for creating menu titles, choice descriptions, Web navigation links, and menu instructions. The standard graphical system conventions inscribed on menus, intent indicators, keyboard equivalents, and keyboard accelerators are also described.

Menu Titles

- **Main menu:**
 - Create a short, simple, clear, and distinctive title, describing the purpose of the entire series of choices.
 - **Submenus:**
 - Submenu titles must be worded exactly the same as the menu choice previously selected to display them.
 - **General:**
 - Locate the title at the top of the listing of choices.
 - Spell out the title fully using either an
 - Uppercase font.
 - Mixed-case font in the sentence or headline style.
 - Superfluous titles may be omitted.
-

A meaningful menu title aids in defining the context of the menu and increases menu comprehension. An experimental study has demonstrated the value of titles to comprehension. Study participants were presented the detailed steps to perform a function. A descriptive title for the steps was (A) not included, (B) presented at the start of the steps, and (C) presented at the end of the steps. Participants given a title at the start of the steps (B), reported higher comprehension and recalled twice as many items as those who were not given a title (A), or who were presented the title at the end (C). The title established the context of the task, and knowing this context greatly aided comprehension.

Main menu. The menu title should immediately orient the viewer to the menu's content and purpose. It should be a short, clear, distinctive, and descriptive title, representing the entire series of choices. It's an important contextual and navigation component. A title such as MAIN MENU OPTIONS provides no information except that the user is probably at the top of a hierarchical menu tree.

Submenus. Submenu titles must be worded exactly the same way as the menu choice previously selected to display them. This will provide structural continuity and assure users that they are progressing as expected through a menu hierarchy.

General. Locate the title at the top of a listing of choices, in the title bar if one is available. Display the title in uppercase or in a mixed-case font using the sentence or headline style of presentation. Whichever style is chosen should be consistently followed for all menus. When using the headline style, capitalize the first letter of each significant title word. The case style chosen should be consistently used on all menus. Superfluous titles, titles that add nothing to the understanding of menu content and context, may be omitted. A pop-up menu requested during a text-editing task, for example, is displayed within the context of the task being performed. The presented choice descriptions by themselves (Copy, Font, and so on) provide the necessary context. Message windows do not need a title either; the text of the message provides the context.

Menu Choice Descriptions

- Create meaningful choice descriptions that are familiar, fully spelled out, concise, and distinctive.
 - Descriptions may be single words, compound words, or multiple words or phrases.
— Exception: Menu bar items should be a single word (if possible).
 - Place the keyword first, usually a verb.
 - Use the sentence or headline style.
 - Use task-oriented not data-oriented wording.
 - Use parallel construction.
 - A menu choice must never have the same wording as its menu title.
 - Identical choices on different menus should be worded identically.
 - Choices should not be numbered.
— Exception: If the listing is numeric in nature, graphic, or a list of varying items, it may be numbered.
 - If menu options will be used in conjunction with a command language, the capitalization and syntax of the choices should be consistent with the command language.
 - Word choices as commands to the computer.
-

Meaningful. Menu item descriptions should be composed of familiar and fully spelled out words. While abbreviations may occasionally be necessary, they should be kept to a minimum. If you are using an abbreviation, only use those that are standard or well known. Descriptions should also be concise, containing as few words as possible, and distinctive, constructed of words that make each choice clearly different from all others. Repeated use of the same word or words in multiple choice descriptions hinders distinctiveness and signals the necessity for creating a grouping whose title is based upon the repeated word.

Use high-imagery keywords, words that elicit a mental image of the object or action. Avoid low-imagery words that have more general connotations. For example, when obtaining a printout of a screen, the term “print” is much more descriptive than “list.”

In creating menu item descriptions, never assume that the description chosen by the designer will have the same meaning to the user. A study has found that the probability of two people choosing the same name or description for something ranged from 8 to 18 percent. Names chosen by experts were no better than those chosen by nonexperts. Therefore, iteratively test and refine the choices to achieve as much agreement in meaning by users as possible.

Size. Item descriptions may be single words, compound words, multiple words, or phrases. Menu bar items should be a single word, if possible. If a menu bar item must be a multiple word, visually tie the two words together by incorporating a hyphen between them. Web page content links will typically be phrases. Link writing guidelines are discussed in more detail in Step 8.

Keyword first. Arrange multi-item descriptions so that the descriptive and unique words appear at its beginning. This optimizes scanning and recognition while the user is learning the menu. Description phrasing and wording should also be consistent across all menus to aid learning further.

Capitalization. Use the headline style of presentation. Whichever style is chosen should be consistently followed for all menus.

Task-oriented wording. Task-oriented wording is preferable to data-oriented wording. Task-oriented wording usually positions a verb first, such as Manage Customer Information. An example of data-oriented wording would be to simply say Customers. What is being done with, for, or to customers is unclear in the latter.

Parallel construction. When choices are composed of phrases, use a parallel word construction in creating descriptions for related choices. Parallel construction would be: *Print* a File, *Execute* a Program, and *Eject* a Disk. An example of non-parallel construction is: *Print*; *Execute* a Program, and *Disk Eject*.

Relationship to title. A menu choice must never have the same wording as the title of the menu on which it is presented.

Consistency across menus. Identical choices on different menus should be worded the same.

Numbering. Items should not be numbered unless the listing is numeric in nature, graphic, or a list of varying items.

Command language. If menu options will be used in conjunction with a command language, the capitalization and syntax of the captions should be consistent with those of the command language.

Word as a command to computer. Phrase all menu choices as commands to the computer whenever possible. For example, say

Choose one:

Save and exit

Exit without saving

rather than

Do you want to save and exit?

Yes

No

Wording a choice as a command to the computer more clearly describes the action of what the command accomplishes. The Yes/No alternatives shown in the preceding example must be comprehended in conjunction with the question being asked. Wording a choice as a command also provides choice phrasing that is consistent with other system commands. A system, for example, often contains the standard commands Save and Exit. In addition, command wording enhances the learning of command mnemonics. Finally, this wording implies that the initiative is with the user in the dialog, not with the computer.

Menu Instructions

- For novice or inexperienced users, provide menu completion instructions.
 - Place the instructions in a position just preceding the part, or parts, of the menu to which they apply.
 - Present instructions in a mixed-case font in sentence style.
 - For expert users, make these instructions easy to ignore by
 - Presenting them in a consistent location.
 - Displaying them in a unique type style and/or color.
-

People not familiar with a system and its menus may need guidance on how to complete a menu. Their needs may, however, have to be balanced against the needs of experienced users who may not want or desire such assistance. To satisfy the needs of all kinds of users at the same time necessitates that menu instructions be included on a menu, but that these instructions be easily ignored by those who do not need them.

Novice or inexperienced users. Provide explicit menu completion instructions for novice or inexperienced menu users. Place the instructions in a position just preceding the part, or parts, of the menu to which they apply. Present the instructions in a mixed-case, sentence-style font.

Expert users. When instructions are included on menus, they must be visually recognized as instructions. This will allow them to be easily ignored by the expert user when they are not needed, or no longer needed. Therefore, some visual aspect of the instruction must indicate that it is an instruction. As mentioned in Step 3, designers of paper forms do this by presenting instructions in a different font or font style such as italics. The form user then immediately recognizes them as instructions, and they can be read or ignored as is desired.

To make instructions immediately recognizable as instructions on a menu, then, present them in a unique font or color. If one of these methods is used, however, cautions concerning the excessive use of different font styles (and colors, as shown in Step 12) must be heeded. Another, but less visually strong, technique is to identify the technique simply by its location. Begin the instruction to the left of the screen elements to which it applies, the left-justification identifying it as an instruction. Guidelines for writing text, including instructions, are discussed in Step 8.

Intent Indicators

- Cascade indicator:
 - To indicate that selection of an item will lead to a submenu, place a triangle or right-pointing solid arrow following the choice.
 - A cascade indicator must designate every cascaded menu.

- To a window indicator:
 - For choices that result in displaying a window to collect more information, place an ellipsis (. . .) immediately following the choice.
 - Exceptions — do not use when an action
 - Causes a warning window to be displayed.
 - May or may not lead to a window.
- Direct action items:
 - For choices that directly perform an action, no special indicator should be placed on the menu.

Providing an indication of what will happen when a menu item is selected can enhance predictability and exploration of a graphical system. If a choice leads to another lower-level menu, include a *cascade indicator*, a right-pointing arrow, following the item description. If a choice leads to a *window*, include an ellipsis following the item description. Items causing a direct action will have no indicator. These intent indicators are illustrated in Figure 4.12.

IBM's SAA CUA designates choices leading to submenus or windows as *routing* choices, and items causing direct actions as *action* choices. A Microsoft Windows *intent indicator* simply implies that additional information is needed. This additional information request is usually presented in a window, but it need not necessarily be restricted to a window.

Keyboard Shortcuts

When a person is using a keyboard, it is inefficient to continually take mouse actions. Keyboard shortcuts allow actions to be taken using the keyboard as well as the mouse. In a study, keyboard shortcuts have been found to be significantly faster and more accurate than mouse clicks, and preferred by nearly all users (Jorgensen et al., 2002). Shortcuts include *keyboard equivalents* and *keyboard accelerators*.

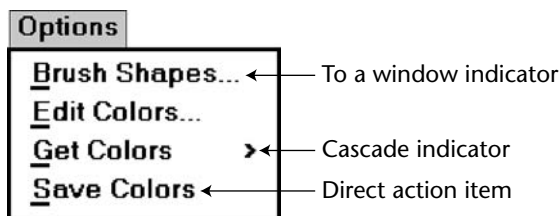


Figure 4.12: Intent indicators.

Keyboard Equivalents

- To facilitate keyboard selection of a menu choice, each menu item should be assigned a keyboard equivalent mnemonic.
- The mnemonic should be the first character of the menu item's description.
 - If duplication exists in first characters, use another character in the duplicated item's description.
 - Preferably choose the first succeeding consonant.
- Designate the mnemonic character by underlining it.
- Use industry-standard keyboard access equivalents when they exist.

Keyboard selection. The ability to select a menu alternative through the keyboard should always be provided. This is accomplished by providing a keyboard *equivalent* for each menu alternative.

Mnemonics. Keyboard equivalents that have meaningful associations with their corresponding choices will be more easily learned and remembered. Studies have found that simple truncation is a good method for creating mnemonics. Therefore, the first letter of the item description is the recommended mnemonic. Unfortunately, in following this method, duplications easily occur, so an alternative principle must also be provided. A simple scheme is to use the second consonant for duplicate items. This duplication-breaking scheme need not always be faithfully followed, however. Occasionally another letter in the menu item may be more meaningful to the user. In these cases, it should be selected.

Designation. Mnemonic codes can be visually indicated in a number of ways. The recommended method is an underline beneath the proper character within the choice. Other methods — a different character color, different character intensity, or a contrasting color bar through the relevant character — are visually more complex and should be avoided. Underlined keyboard equivalents are illustrated in Figure 4.13.

Keyboard equivalent

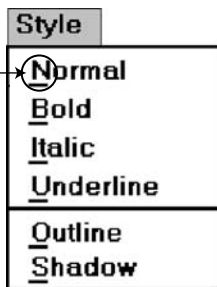


Figure 4.13: Keyboard equivalents.

Table 4.1: Standard Keyboard Equivalents

<u>A</u> bout	<u>H</u> elp	<u>P</u> rint	<u>S</u> end To
<u>A</u> pply	Help <u>T</u> opics	Print <u>P</u> review	<u>S</u> how
<u>B</u> ack	<u>I</u> nsert	<u>P</u> roperties	<u>S</u> ize
<u>B</u> rowse	<u>M</u> aximize	<u>R</u> edo	<u>S</u> plit
<u>C</u> lose	<u>M</u> inimize	<u>R</u> epeat	<u>S</u> top
<u>C</u> opy	<u>M</u> ove	<u>R</u> estore	<u>U</u> ndo
<u>C</u> ut	<u>N</u> ew	<u>R</u> esume	<u>V</u> iew
<u>D</u> elete	<u>N</u> ext	<u>R</u> etry	<u>Y</u> es
<u>E</u> dit	<u>N</u> o	<u>R</u> erun	
<u>E</u> xit	<u>O</u> pen	<u>S</u> ave	
<u>F</u> ile	<u>P</u> aste	<u>S</u> ave <u>A</u> s	
<u>F</u> ind	<u>P</u> age Setup	<u>S</u> elect <u>A</u> ll	

A great deal of commonality exists among these equivalents because they represent a wide variety of functions, many of which will rarely appear together on a single menu. If two actions with the same equivalents will be used within the same menu, one equivalent will have to be modified to make it unique.

Industry standards. Standard industry keyboard equivalents have been established for many common system menu choices. Where these standard equivalents have been established, they should be followed. Microsoft Windows calls keyboard equivalents *access keys*. Standard keyboard equivalents are shown in Table 4.1.

Keyboard Accelerators

- For frequently used items, provide a keyboard accelerator to facilitate keyboard selection.
- The accelerator may be one function key or a combination of keys.
 - Function key shortcuts are easier to learn than modifier plus letter shortcuts.
- Pressing no more than two keys simultaneously is preferred.
 - Do not exceed three simultaneous keystrokes.
- Use a plus (+) sign to indicate that two or more keys must be pressed at the same time.
- Accelerators should have some associative value to the item.
- Identify the keys by their actual key top engraving.
- If keyboard terminology differences exist, use
 - The most common keyboard terminology.
 - Terminology contained on the newest PCs.
- Separate the accelerator from the item description by three spaces.

- Right-align the key descriptions.
- Do not use accelerators for
 - Menu items that have cascaded menus.
 - Pop-up menus.
- Use industry-standard keyboard accelerators when they exist.

Accelerators are keys, or combinations of keys, that invoke an action regardless of cursor or pointer position. They are most commonly used to activate a menu item without opening the menu. They are most useful for frequent activities performed by experienced users. IBM's SAA CUA and Microsoft Windows call these keys *shortcut* keys. They may also be called *hot keys*. Many products have, within their guidelines, standard accelerator key recommendations as well as rules for creating new accelerator keys.

For frequently used items, assign a key, or combination of keys, to accomplish an action. Function key shortcuts are usually easier to learn than modifier plus letter shortcuts. Pressing no more than two keys simultaneously is preferred; three key-strokes is the maximum. Use a plus (+) sign to indicate on the screen menu that two or more keys must be pressed at the same time.

Accelerators should have some associative value to the item and be identified by their actual key-top engraving. In situations where multiple kinds of keyboards exist, and there are keyboard terminology differences, use the term most commonly found on the keyboards or use the term contained on the newest PC, if evolution to the new PCs is expected.

Display the accelerator right-aligned and enclosed in parentheses to the right of the choice. Incorporating these key names within parentheses indicates that they are prompts (which they actually are) and that they may be easily ignored when not being used. Most graphic systems do not place them within parentheses, giving them much too strong a visual emphasis. See Figure 4.14.

Do not use accelerators for menu items that lead to cascaded menus. Also, do not use accelerators on pop-up menus, because they are mouse driven. Use standard keyboard accelerators when they exist. Standard industry accelerators are shown in Table 4.2.

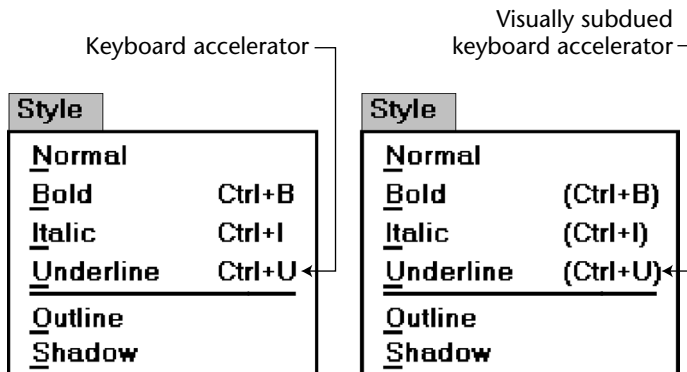


Figure 4.14: Keyboard accelerators.

Table 4.2: Standard Keyboard Accelerators

THIS ACCELERATOR:	DESIGNATES THIS ACTION:
Ctrl+C	Copy.
Ctrl+N	New.
Ctrl+O	Open.
Ctrl+P	Print.
Ctrl+S	Save.
Ctrl+V	Paste.
Ctrl+X	Cut.
Ctrl+Z	Undo.
F1	Display contextual help window.
Shift+F1	Activate context-sensitive help.
Shift+F10	Display pop-up menu.
Spacebar	Select (single mouse click).
Esc	Cancel.
Alt	Activate a menu bar.

Selecting Menu Choices

Menu items can be selected by pointing at the choice with a mechanical pointer, by pointing at the choice through the keyboard, or by keying a value designating the choice.

Initial Cursor Positioning

- If one option has a significantly higher probability of selection, position the cursor at that option.
- If repeating the previously selected option has the highest probability of occurrence, position the cursor at this option.
- If no option has a significantly higher probability of selection, position the cursor at the first option.

When a menu is first displayed, position the cursor at the option most likely to be chosen, or at the first option in the list if no option has a significantly higher probability of selection. If repeating the previously selected option has the highest probability of occurrence, position the cursor at this option.

Choice Selection

- **Pointers:**
 - Select the choice by directly pointing at it with a mechanical device such as a mouse or trackball pointer, or light pen, or pointing with one's finger.
 - Visually indicate
 - Which options can be selected.
 - When the option is directly under the pointer and can be selected.
 - Visually distinguish single- and multiple-choice menu alternatives.
 - If pointing with a mechanical device is the selection method used
 - The selectable target area should be at least twice the size of the active area of the pointing device or displayed pointer. In no case should it be less than 6 millimeters square.
 - Adequate separation must be provided between adjacent target areas.
 - If finger pointing is the selection method used
 - The touch area must be a minimum of 20 to 30 millimeters square.
 - The touch area must encompass the entire caption plus one character around it.
 - **Keyboard:**
 - If moving the cursor to a menu choice
 - The up and down arrow keys should move the cursor up or down vertically oriented menu options.
 - The left and right cursor keys should move the cursor left or right between horizontally oriented menu options.
 - If keying a choice identifier value within an entry field
 - Locate the entry field at the bottom of the last choice in the array of choices.
 - Uppercase, lowercase, and mixed -case typed entries should all be acceptable.
 - **Selection/execution:**
 - Provide separate actions for selecting and executing menu options.
 - Indicate the selected choice through either
 - Highlighting it with a distinctive display technique.
 - Modifying the shape of the cursor.
 - Permit unselecting choice before execution.
 - If a menu is multiple choice, permit all options to be selected before execution.
 - **Combining techniques:**
 - Permit alternative selection techniques, to provide flexibility.
-

Pointers. Items can be selected by being pointed at using a mechanical device such as a mouse, trackball, or light pen, or through touch pointing using one's finger (for touch-sensitive screens). Pressing a key, such as Transmit or Enter, or a mouse button, signals the choice to the computer. Always visually indicate in a distinctive manner which options are selectable and when the option is under the pointer and can be selected. Visually distinguish single- and multiple-choice arrays of menu choices.

An adequate pointing target area should be provided. This area should be at least twice the size of the active area of the displayed pointer of the pointing device. In no case should it be less than 6 millimeters square. To avoid unintended activation of the wrong option, provide adequate separation between selectable areas. Highlighting of the area when selected will also provide indication of an incorrect choice.

If finger pointing is the selection method used, an even larger touch area must be provided, a minimum of 20 to 30 millimeters. Single-character positions on a screen make poor targets for most fingers. Also, keep in mind that using a finger to signify a choice can be taxing on arm muscles, so this approach should only be used in casual or infrequent use situations.

Keyboard. If the user is moving the screen cursor to a menu choice, the up and down arrow keys should move the cursor up and down a vertical column of menu options. The left and right arrow keys should move the cursor left and right across a horizontal array of options. If the user is keying a choice identifier value within an entry field, locate the entry field at the bottom of the last choice in the array of choices. If the user is keying a mnemonic value, the entry should be acceptable in any case (upper, lower, and mixed).

Selection/execution. Provide separate actions for selecting and executing menu options. For example, require typing the mnemonic to select, and then a press of the Enter or Return key to execute. Or, with a mouse, require moving the pointer to the option to select, and then clicking to execute. A study (Chaparro et al., 2000) found that “pointing-and-clicking” rather than a “mouse-over” to open menus on cascading menus takes 18 percent less time, causes fewer errors, and is preferred by users. Always permit erroneous selections to be unselected and, in a multiple-choice menu, permit all options to be selected before execution.

The item selected should be highlighted in some way through a distinctive display technique such as bolding or changing its color. An alternative is to change the shape of the pointer itself. These methods provide direct visual feedback that the proper choice has been selected, reducing the probability of errors in choice selection.

Combining techniques. Permit alternative selection techniques to provide flexibility. If a pointing method is used, also provide a keyboard alternative to accomplish the same task. Pointing will probably be easier for the novice, but many experts prefer the keyboard alternative.

Defaults

- Provide a default whenever possible.
 - Display as bold text.
-

Defaults aid system learning and enhance efficiency. Provide as many as possible. Indicate a default by displaying it in a bold text.

Unavailable Choices

- Unavailable choices should be dimmed or “grayed out.”
 - Do not add or remove items from a menu unless the user takes explicit action to add or remove them through the application.
-

Choices not available to the user should be made visually distinctive by dimming them or graying them out. They must not compete with active items for the user’s attention. Items should not be added or removed from a menu unless the user takes explicit action to do so. Allowing the system to change menu items takes control away from the user and can also lead to user confusion.

Mark Toggles or Settings

- Purpose:
 - Use to designate that an item or feature is active or inactive over a relatively long period of time.
 - Use to provide a reminder that an item or feature is active or inactive.
 - Guidelines:
 - Position the indicator directly to the left of the option.
 - For situations where several nonexclusive choices may be selected, consider including one alternative that deselects all the items and reverts the state to the “normal” condition.
-

Purpose. Mark toggles or settings, illustrated in Figure 4.15, are menu items that toggle between active and not active. When it is active, an indicator is displayed adjacent to the item description. For nonexclusive choices, a check mark is displayed; for mutually exclusive choices, another distinctive symbol, such as a diamond or circle, is displayed. When the item is not active, no mark or symbol will appear.

Examples of items using mark toggles are having a specific application automatically loaded after the system is loaded; having windows automatically reduced to icons when they are made inactive; or making a setting without requiring a dialog box. The purpose of mark toggles is to activate or deactivate an attribute by setting one menu item.

Advantages/disadvantages. Mark toggles provide a visual indication of the state of an item. They are accessed quickly, but may not always be visible. Mark toggles are best suited to items or features that remain active or inactive over relatively long periods of time. They provide good reminders of the state that exists.

Guidelines. Position the mark toggle indicator directly to the left of the menu option. In situations where several nonexclusive choices may be selected on one menu, consider including one alternative that deselects all the items and reverts the state to the normal condition, as illustrated by “Regular” in Figure 4.15.

Regular	F5
✓ B old	Ctrl+B
✓ <i>I</i> talic	Ctrl+I
<u>U</u> nderline	Ctrl+U
^S uperscript	
_S ubscript	
R educe Font	
E nlarge Font	
F _o nts...	

Figure 4.15: Mark toggles or settings.

Toggled Menu Items

- **Purpose:**
 - Use to designate two opposite commands that are accessed frequently.
 - Use when the menu item displayed will clearly indicate that the opposite condition currently exists.
- **Guidelines:**
 - Provide a meaningful, fully spelled-out description of the action.
 - Begin with a verb that unambiguously represents the outcome of the command.
 - Use mixed-case letters, with the first letter of each word capitalized.

Purpose. A toggled menu item is a one-menu item command that toggles back and forth between the current state and its alternative state. When the menu item is first displayed, it reflects the alternative state to the condition that currently exists. For example, in Figure 4.16, if a background grid is currently being displayed, the menu item reads *Hide Grid*. When Hide Grid is selected, the grid is removed from the window, and the menu item dynamically changes to reflect the opposite action. It will now read *Show Grid*. When a grid is again requested, it will change back to Hide Grid. The purpose of toggled menu item is to use a single menu item to designate and activate the one, opposite, alternative of a two-state command setting.

Advantages/disadvantages. Toggled menu items shorten menus, decrease visual clutter, provide quicker access, and foster faster comprehension of the command action. When they are located on a pull-down menu, however, the actions themselves are not always visible. The opposite action reflecting the current state of the attribute, because it too is not visible, can cause uncertainty for novice users concerning what the state actually is. Toggled menu items are also limited in use to commands only.

Guidelines. Use toggled menu items to designate two opposite commands that are accessed frequently. The menu item displayed must be one that clearly indicates that the opposite condition currently exists. The menu captions should clearly state what would happen if the menu item action were requested. It is most meaningful to begin the command with a verb.

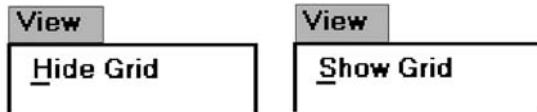


Figure 4.16: Toggled menu item.

Web Site Navigation

Navigation refers to the method people use to find what they want in a Web site. Navigation, and an efficient navigational structure, is the most important element in system usability. A simple and clear navigational structure is the backbone upon which all system features are draped. To navigate to a destination, people use available spatial and environmental information, a process called *wayfinding* (Lynch, 1960; Downs and Stea, 1973). Wayfinding involves four stages: *orientation*, *route decision*, *route monitoring*, and *destination recognition*.

Orientation. One's current location relative to nearby objects and the destination must be determined. This is called orientation. Dividing a space into small parts and providing landmarks and meaningful signage (titles) aids orientation. The result is that locations have identities that are easier to remember.

Route decision. A path must be chosen to get to the destination. Route decision-making is improved by minimizing the number of navigational choices, and by providing signs or prompts at decision points. People generally prefer shorter routes to longer routes, even if the shorter route is more complex. Routes are followed most efficiently when clear directions or signs are available. Site maps provide good mental representations of the space to be navigated, and are very useful if the space is large, complex, or poorly designed.

Route monitoring. The chosen route must be monitored to confirm that it is leading to the proper destination. Connect locations with paths that have clear beginnings, middles, and ends. A person should be able to gauge his or her progress as the path is followed. *Breadcrumbs*, a visible history of the path followed, aids monitoring, especially when a mistake has been made, and going back is necessary.

Destination recognition. The destination, when it is found, must be easily recognized. To aid recognition, provide clear and consistent identities to destinations.

In Web site design, the most successful sites have been found to be those that enhance wayfinding, possessing easy-to-use and understandable navigational systems. No amount of graphical "sparkle" has yet been able to overcome a poor navigational design. A Web site's organizational structure, its navigational tools and their obviousness, and the labels on its navigational elements influence a Web site's usability. The navigational elements of Web sites are referred to as *links*. Navigational elements that link usually consist of textual phrases or images. A person selects a link through clicking a device-based control, such as the mouse, on the link itself. A navigational link, therefore, is said to be *clickable*. A link, when selected or clicked, typically

causes a new page to be presented containing the content described by the link's label. A fundamental principle in Web page design is to clearly visually differentiate clickable links from other page elements.

Other design aspects also influence Web site usability. Among them are the navigational aids available, including indexes. In Web site navigation design, the unique, often incompatible, relationship the browser has to the Web site application being presented can also influence its navigational ease.

Web Site Navigation Problems

To fully understand what comprises good navigation, let's first look at some Web site navigational issues and problems, both technical and usage-related. The Web and its navigation is undoubtedly the most complex computer interface facing people today.

Technical issues. Unlike a graphical system application, whose screens tend to flow in an orderly and predictable manner, a Web site is composed of pages, each of which can, theoretically, be linked to any other page in the site. The graphical application user normally begins a process at a prescribed starting point and proceeds sequentially until a process or task is finished. Web users, on the other hand, can perform tasks or satisfy needs at will, easily moving between most pages in the "spider web" in any order desired, and even jumping to other spider webs when the urge arises. In an analogy to driving a car, the graphical system user is essentially following a freeway in Nevada. The Web site user is wandering around in downtown Boston without a road map and, encountering a road link (a bridge over the Charles River), suddenly finds himself in Cambridge.

The graphical system user must deal with only one operating system whose navigational characteristics are standard and fairly consistent. The Web user must confront two navigational systems, that of the browser being used and that of the Web site being viewed. A click of the browser Back button, for example, simply redisplay the page that was previously displayed. This page may have been in another Web site, and the user is transported there. Neither Web site is aware of this change. Nothing that might have been done on the page "moved back from" is changed. To move around within a Web site requires links within the Web site, either in the form of textual links or command buttons. The data must always be thought of as separate from the controls used to display it, not as being seamless, as occurs in graphical systems. Web users, especially novices, often do not recognize where the browser ends and the Web site begins.

Another problem: Because of the rapidly evolving and expanding nature of the Web, Web sites also have a tendency to grow and grow. As more and more is added, what may have been initially a reasonable structure and menu scheme slowly dissolves into a confusing mass of listings and linked pages. The result is unrelated information that is presented in no particular order.

Usage problems. The two most serious user problems in Web navigation are the heavy mental loads imposed to use the Web and the feeling of spatial disorientation that often occurs. This problem may also occur in hierarchically structured graphical systems. The *cognitive or mental overhead* the user must expend in making decisions concerning which links to follow, or to abandon, can be overwhelming.

Often, there are too many links presented on a page, many of whose meanings are not clear. Links frequently offer few clues to where they lead, how much information will be found at the other end, and how this information relates to the currently displayed page. For the user to reach a goal, each link's relevance to the task at hand must be determined. Another problem is that not all links on a page are always obvious. This often leads to trial-and-error behavior; the user aimlessly clicking to see what happens.

Feelings of *disorientation* are easily experienced when one becomes "lost in Web space." Studies have shown that most people do not seem to become familiar with the layout of sites or develop useful mental models of their structure (Spool et al., 1997). Many people also don't understand where they are in a Web site's information structure (Nielsen, 2000). A scrolling page can lead to loss of local context when the basic navigational elements, such as links to other local pages in the Web site, disappear. There are then no familiar landmarks to then navigate by. Long chains of links to reach relevant material can be tedious and also lead to loss of context, and a "Where am I?" reaction.

A research paper found that 39 percent of users of shopping sites failed in their buying attempts because the sites were too difficult to navigate. If people get buried in information, or lost on a side trip with no signposts or landmarks in sight, the most frequently implemented solution to the problem is to abandon the entire process.

Web Site Navigation Goals

As previously described, understanding a Web site's navigational scheme is made more difficult because Web sites usually have much less perceived structure than typical graphical system applications. Web pages can also be of any length and possess any number of links to any number of other pages. The user can wander at will or whim through multitudes of links, pages, and Web sites, and any meaningful structure a Web site design does possess can easily disappear from one's memory in the maze of directional twists and turns being made. The potential for getting lost is extremely high, unless numerous, obvious, and understandable landmarks are available as a guide.

MAXIM The time it takes to make a decision increases as the number of choices increases.

A well-designed navigation system facilitates quick and easy navigation between components whose structure and relationship are easily comprehensible. For the user, answers to the following questions must be obvious at all times during an interaction:

- Where am I now?
- Where did I come from?
- Where can I go from here?
- How can I get there quickly?

A good navigational scheme and features, and the proper navigational tools, will minimize, if not eliminate, the problems associated with cognitive or mental overload and feelings of disorientation.

Web Site Navigation Design

The section focuses specifically on Web site navigation design. It will review typical Web site organizational schemes; the navigational components of a site, including the browser command buttons, links, Web site toolbars, and Web site command buttons; and the characteristics and components of a Web site that contribute to maintaining a “sense of place.” In designing a Web site navigation scheme there are two things to always remember. Never assume that users know as much about a site as the site designers do (this has been said before), and any page can be an entry point into the Web site.

Web Site Navigation Aids

- To aid Web site navigation and learning
 - Provide a map or overview of the menu hierarchy.
 - Provide clickability cues.
 - Provide a “look ahead” at the next level of choices, alternatives that will be presented when a currently viewed choice is selected.
 - Change the color of a link that has been clicked.
 - Provide feedback concerning one’s current location
 - Provide navigation history.
 - Match link text (or label) to the destination page heading.
-

Map or overview. As has been discussed, as one wanders deeper into a multilevel menu system, it is increasingly difficult to maintain a sense of position or orientation. The result is that getting lost in the maze is quite easy to do. The value of a map or overview in reducing disorientation has been demonstrated in some studies. In these studies, providing a graphic representation of a menu structure in map form, either in hard copy or online, resulted in fewer errors or wrong choices, faster navigation, and greater user satisfaction when compared to providing no guides or simply providing indexes or narrative descriptions of the menu structure. So, maps or graphic representations of the Web site structure are desirable and should be included on the Homepage. They should also be included in the Web site documentation (where available), and through a Help function.

Clickability cues. It should be obvious which items on a page are clickable. Provide a visual indication that an item or word on a page is clickable using techniques such as color, underlining, bullets, and arrows.

Look-aheads. Navigation and learning will be assisted if a person is able to browse the next level of choices before the currently displayed choice is selected. As the

cursor moves across a menu bar, for example, the pull-down menu may be automatically dropped, permitting review of the choices available if that menu bar item is selected. Such look-aheads are useful if ambiguity exists at higher-level choice points. They have been found to decrease errors and improve satisfaction. Menu search time may be longer, however.

Link color. Changing the color of a link that has been selected reminds a person that this destination has already been visited.

Navigation history. It has been found that being able to view, on the screen, the path one is following improves learning and performance, and reduces feelings of disorientation. Provide a navigation history that summarizes the menu choices made leading to the currently displayed menu or screen.

Link text. The importance of good Web site labeling has already been discussed. Matching link text and destination page heading will assure the user that the path being followed is correct.

Web Site Organization

- Divide content into logical fragments, units, or chunks.
 - Establish a hierarchy of generality or importance.
 - Structure the relationships among content fragments, units, or chunks.
 - Establish global or site-wide navigation requirements.
 - Create a well-balanced hierarchical tree.
 - Restrict to two levels requiring no more than two clicks to reach deepest content, whenever possible.
-

It is easier to develop a clear and comprehensible navigation scheme if the Web site is organized and structured in a meaningful way. The design goal is a proper balance of menus and pages that can be easily and efficiently moved between.

Logical fragments, units, or chunks. Because of limitations in short-term human memory, smaller discrete fragments or chunks of information are often easier to navigate than long, undifferentiated units. The concept employed in Web site design, in reaction to this human memory frailty, is called *hypertext*. Hypertext is a nonlinear way of organizing information based upon the following principles:

- A large body of information exists that can be organized into fragments.
- The fragments relate to one another.
- The user needs only a small fraction of the fragments at any one time.

In organizing a Web site, information is first divided into logical fragments, units, or chunks. Coherent chunks that focus on a single topic is the desired goal. These small chunks of related information are easier to organize into the modular groups of information that will compose the organization scheme, and form the basis for hypertext links to be described shortly. A design-organizing aid, card sorting, is discussed in Step 2.

Hierarchy of generality or importance. Having identified the information units, information is now organized in according to importance or generality, from general to specific. A hierarchical tree is the most recommended organization scheme; Sun Microsystems (1998) suggests that whenever possible

- State conclusions and link to supporting details.
- Enumerate categories of information and link them to detailed listings.
- Summarize information and link to full-length treatments.

A document organizational tree structure, (table of contents, chapters, sections, and subsections) is a good scheme, because people are very familiar with, and have an excellent mental model of this organization. Such a structure provides information about information sequence, information quantity, and the relationships existing between components. Other organizational schemes include topics followed by subtopics, or prioritization from most to least important. The objective is to allow the user to scan the page and then select relevant and useful content for further review. Excessive fragmentation of a long, sequential story, however, should be avoided. Reading will be impeded and printing made more difficult.

Structure the relationships. Identify the relationships that exist between various elements in the hierarchical tree. In a large Web site, two levels of navigation will exist. The first is movement within the subject area. This navigation includes moving *within a branch*—up to a parent page or down to a child page. It also involves navigating across branches to sibling pages or other sections of a site. What points on other tree branches it will be beneficial to go directly to then, must also be established. The second navigation type is *global or site-wide*. What other site features, such as search a facility, site maps, and other major content areas should be mentioned on each page? Do not mention all features on all pages. Restrict the number presented to the several most useful features.

To unveil the Web site's structure, use *progressive disclosure*. Heading levels, shown in varying type sizes (as on paper), will also be helpful in aiding understanding of site organization.

Hierarchical tree. Web site pages should be organized as offshoots of a single homepage. If a site has a large number of information categories, and each category contains a lot of content, create submenus to aid navigation. The design goal: a well-balanced hierarchical tree that facilitates quick access to all information and also helps people understand how the site is organized. The so-called spoke design, where every page is linked to every other page, has been found to lead to lower usability.

Hierarchical breadth has been found by many research studies to be greatly preferable to hierarchy depth. A few menus with a larger number of choices are better than a large number of menus each with a smaller amount of choices. When menu levels go to four, five, or more, the chance of users becoming lost or disoriented is greatly increased. As studies have found (Zaphiris and Mtei, 1998; Larson and Czerwinski, 1998), restrict, whenever possible, the hierarchical tree to two levels requiring no more than two clicks to reach the deepest content. A two-level structure encompasses a homepage and two additional levels below it.

Navigation Page Design

- Use appropriate menu types.
 - Sequential menus for simple forward-moving tasks.
 - Simultaneous menus for tasks that would otherwise require extensive Back button use.
 - Keep navigation-only pages short.
 - Limit prose text.
 - Scrolling:
 - Never require scrolling of navigation-only pages.
 - Minimize the need for scrolling to view all links on pages containing content.
 - Never require horizontal scrolling.
-

A navigation page contains no content and is designed to direct or redirect users. It may take the form of a homepage, a site map, an overview, and so forth.

Appropriate menu types. Use *sequential* menus, menus arranged in a predetermined order, for simple forward-moving tasks. Use *simultaneous* menus, menus displayed together, if the use of sequential menus will require extensive use of the Back button (Hochheiser and Shneiderman, 2000). Simultaneous menus are usually presented in frames, a feature supported by most browsers. A frame allows the display area to be divided into two or more sections whose contents behave like independent Web pages.

Short pages. Confine navigation pages to one screen whenever possible. Nonvisible items may never be noticed.

Prose text. When navigation pages contain many words, readers tend to scan looking for specific words and clicking links rather than reading the text associated with links (Koyani, 2004.)

Scrolling. Never require scrolling of navigation-only pages. Besides being tedious, not being able to see all links at the same time makes comparison of the alternatives for selection purposes much more difficult. For scrollable content pages, minimize the need to scroll to see all links. Also, ensure that all related links on a screen are seen together to facilitate comparison. Never require horizontal scrolling. It makes text reading difficult and users dislike it.

Components of a Web Navigation System

To move between Web site information fragments necessitates the creation of many navigation *links*. They are contained within a framework of tools or controls, including the browser's command buttons, textual phrases, Web site navigation bars, and Web site command buttons. Collectively, these are all referred to as links. Links are one of the most discussed issues in Web site design.

A link functions as a menu choice that, when selected, results in the connected information being displayed, or results in a file being opened or downloaded. A movement

link may transport the user to another location within a page, to a new site page, or to another Web site. Originally, due to the nature of technology at the time hypertext was employed in computer systems, links only consisted of textual or binary files. Utilization of hypertext on the Web allowed links to be created using images as well as text, so the term *hypermedia* was coined to reflect this expanded nature.

In addition to being the critical component in Web navigation, links give the user an idea of what a Web site, document, or page is all about. The wording of a textual link should enable a person to predict what lies submerged below, or what will happen if it is activated. Descriptive links let the user determine whether a link should be followed or not. This is a complex cost-benefit calculation that the user makes many times in a Web interaction session.

Providing an extensive collection of link navigation tools will focus the user on the Web site itself and its content, drawing attention away from the general-purpose browser links. Making these tools consistent and predictable will help the user create an understandable mental model of the site and its organization. To begin, several general link guidelines are

- All navigation elements must
 - Make sense in the absence of site context.
 - Be continually available.
 - Be obvious and distinctive.
 - Be consistent in appearance, function, and ordering.
 - Possess a textual label or description.
 - Offer multiple navigation paths.

Sensible. All navigation controls, in the absence of site context, must make sense to the user. The user may have “lost” the context, or the page or Web site may have been entered from almost anywhere.

Available. All navigational controls must be easy to access. If they are not readily available, the full advantages of hypermedia may not be achieved.

Obvious and distinctive. A navigation link or control must look like a navigation control. Its appearance to the user must immediately suggest that it is an entity to be clicked or otherwise selected. This is accomplished through a control’s appearance as well as its location. Non-obvious link or control choices lead to aimless and tedious page clicking and ultimately confusion and frustration. Conversely, do not make any other screen element look like a navigation tool if it is not one.

The obviousness of a link is called its *affordance*. A control with high affordance will be quickly identified as a control. Bailey (2000) in a study compared the link affordances of the homepages for two large Web sites. Each page contained 29 links. The link affordance rate for one site was 97 percent (participants, on the average, identified 28.2 page links). The rate for the other site was only 76 percent (the average link identification rate being 21.9 per page). This difference was statistically significant. Because of the non-obviousness of one-quarter of the poorer site’s links, its users would have spent longer times searching for links, and would probably not have even discovered some links. Techniques to create the necessary affordance and distinctiveness differ depending upon the kind of

link. Guidelines enabling the various controls to achieve distinctiveness are described in the following control-specific sections.

Consistent. Like all elements of the interface, navigation links, toolbars, and command buttons must be consistent in appearance and behavior.

Textual. All navigation must have a textual label or description. Navigation using textual descriptions is much preferable to graphical-only navigation because the purpose and function of graphic images are often unclear. They also take longer to download. Textual links are also necessary for users who do not have graphics, or who have chosen not to display graphics.

Provide multiple navigation paths. Offer multiple paths or ways to move around the Web. Provide structural components such as site maps, a table of contents, and indexes to go directly to a point of interest, provide content links to move around nonsequentially, and provide command buttons, such as Next and Previous, to move sequentially.

Navigation Elements

- Differentiate and group navigation elements.
 - Provide a global navigation bar at the top of each page.
 - Provide a local category or topical links navigation bar on the left side of a page.
 - For long lists, consider placing within a frame.
 - Optionally, provide a secondary navigation column on the right side of the page.
 - Provide explicit or embedded textual links within the contents area.
 - Consider duplicating embedded links in the left side navigation bar.
 - Place minor illustrative, parenthetical, or footnote links at the end of the page.
 - For long pages provide
 - “List of Content” Links.
 - Important global or local links in a navigation bar repeated at the page bottom.
 - Create a common and consistent theme.
 - Never create pages without navigational options.
-

A Web site contains at least three levels of navigation elements: *global* or *site-wide*; *local* and *specific*; and *minor* or *footnote*. Clearly differentiate these navigation elements from one another and locate them consistently from page to page. The recommended structure, illustrated in Figure 4.17, separates these navigation elements from content, making it easy for users to find each. People using Web sites are now becoming accustomed to finding important navigation elements at the page top and in panels on the left and right side.

Two studies looking at user performance in using, and preferences for, the location of Web site navigational options have been described by Bailey (2006). Alternatives reviewed included options located at the page top [T], down the left side [L], and down the right side [R]. The study researchers, Kingsburg and Andre (2004), began by surveying existing Web site structures. They reported that the most common navigational structures were: (1) An initial selection is made from a page top navigation panel followed by a selection, and subsequent selections, from the left navigation panel (TLL), and (2) initial and subsequent selections are all made from the left panel (LLL).

In a scenario requiring three navigational selections, their first study evaluated top and left panels only. In general, they found

- Navigation was faster
 - When the first and second selection was made from the left panel.
 - When the panel used for the first selection (top or left) was split or separated from the panels used for the second and third selections.
- People preferred
 - The first selection be made from the left panel.
 - The first and second selections both be made from same panel, top or left.
 - The second and third, or first, second and third selections be made from the same panel.
- The best performing and most preferred structures were
 - Left-left-left (LLL).
 - Left-top-top (LTT).

Next, Kingsburg and Andre added a right-side panel in a similarly structured study. They found

- Navigation was faster
 - When the first selection was made from the left panel (not top or right).
 - When all selections were made from the same panel.
- People preferred
 - When all selections were made from the same panel.

Among this study's findings were that a right panel is a viable design option. Overall conclusions were

- Selection limited to either the left or right panels resulted in best performance and was preferred by users.
- Performance-wise, it is better to start in the left, not the right panel.

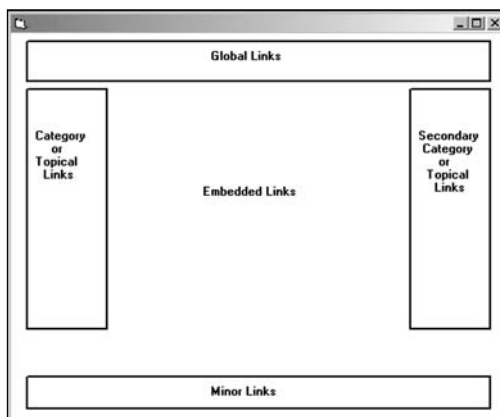


Figure 4.17: Web navigation component locations.

The navigation structures yielding slower performance and lower preference ratings were Top-Top-Top (TTT), Top-Left-Top (TLT), and Right-Top-Right (RTR). Perhaps these alternatives came out poorly because of less efficient scanning organization (TTT), excessive eye and pointer movement (TLT, RTR), or backward eye flow (RTR).

Another recent study by Oulasvirta et al. (2004) found that for people reading left-to-right languages, the tendency is to look to the left for the navigation panel.

Global. Global or site-wide navigation elements provide access to the site's total scope or categories of available information. An evolving standard in design is to locate the global navigation elements horizontally at a page's top. Locating the global links at the page top makes sense if one considers the logical flow of information through a screen. A selection from this global area eventually results in display of a page and its content, a top-to-bottom sequential eye flow. In the eye-tracking study reported by Nielsen (2006) in Step 3, a user's first search of a Web page horizontally across the page top (top bar in the F) may reflect an expectancy that important navigational elements are across the page top.

Category or topical. Local, specific and contextual navigation elements within the category or topical area being presented are typically displayed in a columnar array down the left page side. For long lists consider placing the links within a frame navigation panel. A study found users preferred non-scrollable frames rather than having the links move off as a page is scrolled (Bernard et al. 2001d). A second listing of links can also be presented in a column on the right side. Again, in the eye-tracking study reported by Nielsen (2006) in Step 3, a user's early vertical search of a Web page's left side (vertical bar in the F) may reflect an expectancy that important navigational elements are also along the left side of the page.

Embedded links. Phrases or embedded links will be provided within the contents area of a Web page. An *embedded* link is one found in the middle of prose or continuous text. Embedded links are frequently used to lead to supporting information or provide definitions of terms. They are designate by an underline and a unique color. Because users preferred redundant links, consider duplicating embedded links in the left side navigation bar (Bernard et al., 2001d).

Minor. Minor illustrative, parenthetical, or footnote links can be arrayed horizontally at the page bottom.

List of Content. For long pages with sections that are not visible without page scrolling include a set of links to each page section at the top of the page. These "anchor" or "within page" links provide a reminder of the page's contents, a page outline that can easily be reviewed, and a quick way to navigate to desired sections. These links also assist people in getting to a specific section if they arrive from a different page.

Important links. For long scrolling pages, repeat important global or local links at the page bottom. When finishing a page, the user, then, will not have to scroll upward to locate important navigation links.

Common theme. A common and consistent Web site navigation theme will enable people to more easily understand and learn its structure. Incorporate different

styles for these different navigation elements to aid people in understanding the differences in their meaning and function.

Always present options. All pages must have navigation options. Never create pages without navigation options. Many Web pages contain links opening a new browser window, thereby disabling the browser Back button. If the new window opens in a full screen people may not realize that they have been directed to another window and they may not know what to do. Links that do not behave as expected inhibit a person's understanding of a system. If such links are incorporated within a Web site, always include a prominent action control on the new window to close it and return the user to the original window.

Other Web Site Navigation Elements

In addition to Navigation bars, many other Web site elements are important components of the Web navigation system. Among these are overviews, including executive summaries, site maps, indexes, and tables of contents. Other elements are historical trails and search engines.

Overviews

- Provide
 - An executive summary that provides a preview of the site and contains links to all major concepts.
 - A site map illustrating the site's hierarchical structure and the relationships of components.
 - Both global and local maps.
 - An alphabetized site index.
 - A table of contents.
 - Allow accessibility from any point in the Web site.
-

Overviews provide a top-level view of a site's organization and content. Having an understanding of how a site is organized, the landmarks available within it, and the content it contains, assists the navigation process. In driving an automobile, referring to a road map before embarking on trip usually results in reaching one's destination faster, easier recovery from inadvertent wrong turns, a better ability to handle any unexpected detours that may be encountered, and a less stressful trip.

Overviews are most useful if provided in several forms. They may be needed during a Web interaction as well as before starting into a site. A graphical system help function, for example, may be available in tutorial form, be accessible by topics, or be organized in alphabetic form for easy scanning. It is difficult to predict the user's exact need at any moment in a session.

An *executive summary* will provide an overview of the site in narrative form and contain links to all major concepts. For large Web sites, a *site map* can be used to illustrate the site's hierarchical structure in either graphical or textual form. These elements provide a prospective on one's position in the spatial hierarchy. A good site map helps

facilitate site learning and should encourage comprehensive exploration of a site. Maps may be made available at both a global or local level within the site, depending upon the site's complexity. They can be designed to resemble a traditional table of contents or a simple index. An *alphabetized site index* will permit quick access through keywords and specific topics. A *table of contents*, structured as in a printed book, will permit review of major topics and the subtopics within. Because, based upon user studies, site maps are not always obvious and easy to find, a clear link saying *Site Map* should be placed on a consistent location on every page. A site map should be presented in one page, if possible. Do not exceed a couple of screenfuls, however. All of a Web site's overview elements should be accessible from any point within the site.

Historical Trails

- Provide
 - Breadcrumb trails.
 - Locate at the top of the page below the navigation links.
 - History lists.
 - History trees.
 - Footprints.
 - Bookmarks.
-

Historical navigation aids try to show the user's position in an information space by showing where they have come from, or where they have been. Seeing a navigation path is thought to enable a user to better understand the context of the currently displayed page. Displayed paths also provide a means to easily return to places of interest.

Breadcrumb trail. A *breadcrumb trail* in a hierarchical Web site structure is a sequential textual listing of pages traversed from the parent page to the page currently being displayed. A trail, illustrated in Figure 4.18, is also a series of links that permit the user to go back to any page in the sequence with one click. Breadcrumb trails normally appear near the top of a page.

Studies have shown that spontaneous breadcrumb usage is not high in Web site navigation. The question being asked is, Why? Is it because they are not noticeable on a page, is it because people don't know what they are, or is because people don't care about them? Studies exploring their value have had mixed results. Some studies have found more efficient navigation and/or improved user satisfaction (Bowler et al., 2001; Maldonado and Resnick, 2002; Hull et al., 2004). Another found no navigation benefits (Lida et al., 2003). So, the jury remains out on their utility. Perhaps, like scrolling, people will learn to use breadcrumbs. Because they do provide some users some benefits now, and perhaps usage will increase with user experience, their inclusion on a page still appears beneficial.

At this moment no standard exists for how to separate the page names in a trail. Symbols used include an arrow (->), a colon (:), a greater than sign (>), and a slash (/). Until a standard evolves, any of the above symbols remain acceptable. Do not use anything else, however. Position a breadcrumb trail at the top of the page below any existing navigation links.

[useit.com](#) → [Papers and Essays](#) → [Heuristic Evaluation](#) → List of Heuristics

IBM developerWorks : [Web architecture](#) : [Web architecture articles](#)

[Weather](#) > [Pacific Rim](#) > [Australia](#) > [Sydney](#)

Figure 4.18: Breadcrumb trails.

Other historical trail elements. A *history list* is a sequential textual listing of sites or pages visited over a specific time period, a session, a day, or some other time period. A *history tree* is an overview map of a site's structure with pages already visited marked by an indicator such as a plus sign, check mark, or asterisk. The markings serve as *footprints*, guiding the user back to pages of interest, and/or signaling which have already been seen and may no longer be of interest. A *bookmark* is similar to a history list except that it is designated by the user to mark locations of continuing interest.

Search Facility

- Provide a search facility.

Another form of navigation support is provided by a site search facility. Provide such a facility within larger sites. Search facilities were addressed in Step 3.

MYTH Real users don't mind complex navigation.

Links and Actions

A Web page consists of a collection of links, both textual and graphic in nature, and a sprinkling of toolbars and command buttons. Links are commonly used to go to information, usually on another page. Commands and toolbars are commonly used to perform actions. How should links be presented to make them obvious? What kinds of links should be included on a page? How many links should exist on a page? These and similar questions are addressed next.

Kinds of Links

Typically, three kinds of links are provided for Web sites: internal, anchor, and external.

Internal links provide navigation within a Web site, permitting the user to freely move about between a site's pages. *Anchor* links, also called *associative* or *within-page links*, are used when a page is exceptionally long. A page contents list is presented at the top of the page with links to the corresponding information or section within the page. When clicked, the corresponding section is then displayed. *External* links point to new pages on other Web sites.

Navigational elements consist of textual phrases, images, and command buttons.

Textual Phrase Links

- Provide a mix of textual phrase links
 - In explicit menus.
 - Embedded within page text.

Textual phrases are words, or short pieces of highlighted text, serving as links. Text links are the preferred style of link in Web page design because they are more easily recognizable as clickable, download faster, are more understandable than images, and are preferred by users. They can also be easily modified visually to indicate that they have already been clicked. Another advantage is that using text links enables people with text-only and deactivated graphical browsers to see the navigation options.

Textual phrase links possess two distinct structures: explicit and embedded. An *explicit* menu is a listing of textual phrase links set apart from the main page content, often in toolbars or panels. These listings usually include links to various Web site topics, links to site global features such as the site map or search facility, and perhaps links to other related sites. These listings closely resemble typical screen menu arrays in their structure and presentation. A typical explicit menu is shown in Figure 4.19. An *embedded* menu is a link contained within the textual content of a page. Certain words or phrases are designated as links, highlighted, and when selected display the linked component for the user. An embedded menu is illustrated in Figure 4.20. Web sites usually contain both explicit link listings and embedded links in various mixes.

Lai and Waugh (1995) studied the effect of explicit listing hierarchical menus and embedded menus on a search task. They compared the three menu structures: (1) menus composed solely of explicit listings, (2) solely embedded menus, and (3) mixed explicit and embedded menus. They found that the best menu structure was determined by the kind of search task performed. Explicit listings worked better for straightforward search tasks, whereas menus containing embedded links worked best for complex and not fully known searches. The embedded menus improved search accuracy, but not search efficiency.

Bernard and Hull (2002) found no reliable performance differences between embedded links and explicit links. Users, however, preferred to have them embedded. Bernard et al., (2001d) found that redundant links were preferred by users, that is, duplicate links that were both embedded and outside the text on the left margin. Providing a mix of explicit listings and embedded menus in Web site design will best satisfy the needs of all site users.

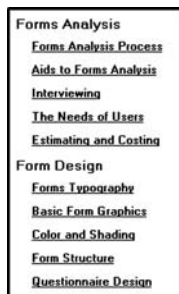


Figure 4.19: Textual explicit links.

the lists of usability problems found by heuristic evaluation will tend to be dominated by minor problems, which is one reason severity ratings form a useful supplement to the method. Even though major usability problems are by definition the most important ones to find and to fix, minor usability problems are still

Figure 4.20: Textual embedded link.

Image Links

Graphical images or icons may appear in an array in the form of a navigation bar, as illustrated in Figure 4.21, or be individually located at relevant points within a page. As just described, text links are more easily recognizable than images, even if the images contain textual phrases. In one study people showed considerable confusion concerning whether or not certain page images were clickable (Koyani et al., 2004). People could not tell whether the images were true links without placing the pointer over the images. This is a slow process. Guidelines for creating and displaying graphics and icons are discussed in Step 11.

Command Buttons and Toolbars

Command buttons and toolbars, used to perform actions, may appear in an array in the form of a navigation bar, or be individually located at relevant points in a page. The advantage of standard Windows-type graphical command buttons are that many people commonly recognize them as clickable elements. This may not be true for unique and more stylized buttons, however. Command buttons and toolbars should never be used to retrieve or show information. Always use links for this purpose. Guidelines for creating and displaying command buttons and toolbars are discussed in Step 7.



Figure 4.21: Graphical or iconic navigation bars.



Figure 4.22: Command button navigation bar.

Presenting Links

- Provide consistent clickability cues.
 - Avoid misleading cues to click.
 - Use Textual links.
 - Underline all link text, including that
 - Embedded in page content.
 - Contained in explicit menu listings.
 - Contained in headings.
 - Used as graphical labels.
 - Exceptions:
 - Links on navigation-only menus and in lists do not necessarily need underlining.
 - Do not underline any text that is not a link.
 - Distinguish between unselected/unvisited links and selected/visited links.
 - Make unselected /unvisited links blue.
 - Make selected/visited links purple.
 - Never show other text in the chosen selected/unselected colors.
 - Distinguish internal, external, and anchor links.
 - Identify external links by
 - Including destination URL address below the link.
 - Including an “exit disclaimer” adjacent to the link.
 - Providing an interim page after clicking an external link.
 - Identify an anchor link with a “Page Contents” heading.
 - Graphical links:
 - Clearly identify clickable regions of images.
 - Distinguish graphical links from decorative graphics through underlining graphical text labels.
 - Links in toolbars:
 - Distinguish links contained in toolbars through:
 - Presenting in consistent locations.
 - Using different colored backgrounds.
 - Fat links:
 - Consider fat links, if appropriate.
-

Links must be easy to find. They must not be confused with other screen graphics or textual content. Having to search for links can be a tedious and frustrating process. Whether a link has been navigated before must also be obvious. When looking for something new, continually embarking down a path already traveled can also be frustrating.

Clickability cues. It should be obvious which items on a page are clickable. Provide a visual indication that an item or word on a page is clickable using techniques such as color, underlining, bullets, and arrows. People should not have to move the cursor around a Web site to determine what is clickable. Scanning a page with a cursor is called *minesweeping*. Scanning with one's eyes is much faster than minesweeping. Be consistent in the use of conventions and techniques to indicate

links. Use of a particular symbol should (1) always indicate clickability, or (2) never indicate clickability. Otherwise the user will be confused and a mental model of the system will be more difficult to learn.

Underline text links. To identify a link, the well-established convention is to underline the link text. All link text must be underlined, including that embedded in page content, that presented in explicit listings, that contained in headings, and that taking the form of labels in graphical images.

If design clearly indicates a page area's navigational function, underlining may be safely eliminated. Consider, however, displaying an underline under non-underlined links when the pointer is placed over the text. This will reinforce the element's clickability.

Designate used links. Unselected or unvisited links must be distinguishable from selected or visited links. The ability to understand what links have been followed is one of the few standard navigational aids available in browsers. In a study, providing this type of feedback was the only variable that increased a person's speed in finding information (Koyani et al., (2004). Stick with the default colors of blue for links already followed and purple for links not yet ventured down. While the choice of blue as a text color was poor because of its degraded reading ability, it is now well learned. Van Schaik and Ling (2003) found that blue links are easier to click than black ones, even though black ones have higher visual contrast and are easier to see. Blue's use is recommended because it is now very familiar. Using nonstandard link colors can lead to problems. It is difficult to remember which color means what, thereby increasing link selection errors. It can also lead to confusion with normal underlined text in a document. If on a Web site multiple links lead to the same destination, change all links to the "visited" color. Finally, to avoid viewer confusion, never present other page components in a link's chosen selected/unselected colors.

Distinguish kinds of links. Visually distinguish links leading to different Web destinations. A study found that people often assume that a link will take them to another page in the Web site where they are browsing (Koyani et al., 2004). If the link simply moves within a page, or goes to another Web site entirely, confusion can result. A link that moves to another Web site can also be aggravating if the person was not ready to move on. A link's destination should be as predictable as the content at the other end.

Koyani et al. recommend the following methods for notifying a user that a link will leave the current Web site. Include the *URL address* of the destination below the link text as shown in Figure 4.23. Provide an *Exit disclaimer* command button adjacent to the link as shown in Figure 4.24. Provide an *interim page* after clicking the external link and before going to the new Web site as illustrated in Figure 4.25. Whichever technique is chosen, it should be followed consistently within a Web site.

Designation of these different destination types can also be accomplished by grouping links by type, giving them a descriptive heading, and placing them in unique and consistent locations on a page. Anchor links, for example, can be given a heading such as "Page Contents" as illustrated in Figure 4.26. A destination convention has yet to be established in Web site design. When one is established, it should be applied.



Figure 4.23: Destination URL address.



Figure 4.24: Web site exit disclaimer.



Figure 4.25: Web site interim page.



Figure 4.26: Anchor links table of contents.

Graphical links. Clearly identify clickable regions of images, either the entire image or the relevant sections of the image. Users should not have to use the mouse pointer to locate the clickable area or areas. If care is not exercised, graphical links may be also confused with decorative graphics. A principle of graphical or icon design is to always provide a text label (see Step 11). Distinguish graphical links by underlining the graphical text labels, as is done with plain link text.

Links in toolbars. Distinguish links contained in toolbars from page content by presenting the toolbars in consistent locations, and/or displaying them in backgrounds of a contrasting color to the page content. Global toolbars, most often in iconic or button form, are becoming consistently arrayed across the top of a page. Category or topical toolbars are now commonly arrayed down the left side or right side of the page. Toolbars containing textual listings that are similar to page content can be emphasized and differentiated through presenting them with a background and style that contrasts with the content.

Table 4.3: Links to Avoid (or Links that Aggravate the User)

Orphan Link	A link leading to a page that does not possess any navigation options.
Boomerang Link	A links that returns to the exact same spot.
Gotcha Link	A link that leads to little or no content.
False Alarm Link	A warning to not follow a link you really should follow.
Mystery Link	A link that does not look like a link because it is not properly labeled or does not possess a raised appearance.
Link-mania	Linking every time the same keyword is mentioned in a page.
Link-drunk	A long succession of links that must be followed to reach the destination.
Stairmaster Links	No Next link in a series of pages, necessitating continual return to a table of contents.
Gratuitous Link	A link to other sites to return a favor.
Missed Opportunities	For useful links.

Fat links. Links pointing to more than one page are called fat links. Because the browsers Firefox and Safari support tabbed browsing, it is possible to have a link open up into multiple tabs. Several destinations can then be accessed at once.

Some kinds of links to avoid are summarized in Table 4.3.

Types of Links

- Internal links within a page:
 - For long pages, include anchor links to internal page content.
- Internal links within a Web site:
 - On all pages include links to
 - The Web site homepage.
 - Global Web site features.
 - Other main pages, navigation points, or categories.
 - The likely Web site starting point.
 - Main pages with links to the displayed page.
 - On sequential pages, include links to the
 - Next page.
 - Previous page.
 - Also consider including links to
 - Places of related interest.
 - Important pages.
 - Background or explanatory information.
 - Supplemental information.

- New or changed content.
 - Web site Quit or Exit.
 - Repeat important links.
 - External links:
 - Most appropriate for informational sites.
 - Provide links to relevant information on other Web sites.
 - Related content.
 - Reference information.
 - Background reading.
 - Place external links on a separate page.
 - Provide an indication when a link goes outside the current site.
-

Internally within a page. For long Web site pages, include *anchor* links for important content within the page. Place these links at the top of the page and identify them by a heading as anchor links.

Internally *within a Web site*, on all pages include the following links:

Homepage. A home link will transport the user directly to the site's homepage, a stable and safe starting point to escape to in times of difficulty. Easy access is also achieved when the user is ready to start over, or ready to commence a new navigation. A home link eliminates the necessity for sequential backward movement up through a series of pages.

Global features. Provide links to a site's global features, including the highest level of information categories and utilities such as the Search facility.

Other main pages, navigation points, sections, or categories. Do not link to all sections of the site from all pages; to do so will be overwhelming. To provide easy navigation throughout a site, provide links to a site's major navigation points, sections, or categories of information. Pages linked to must however, possess substantive content.

MYTH Why do users need a road map of a Web site? They'll know where to go.

The likely Web site starting point. Provide links to the site's likely starting point, the homepage, a site map, or an index.

Main pages with links to page. Provide links back to the main pages that have links to the displayed page. A return link describing the page one is going back to provides better predictability and much clearer context. It also provides escapability. While the browser's Back button will accomplish the same thing, it does not say what it is going back too, in case the user has forgotten where arrival was from. It also keeps navigation within the application itself (as opposed to the browser).

For *sequential pages*, provide easily accessible links to adjacent pages.

Next. To allow sequential movement downward through pages, place a Next link at the end of each page. Explain, whenever possible, what will happen or where one will go when the link is selected. Without this link, the user will have to continually refer to a table of contents or menu listing to continue navigation. This link will also allow users, should they choose, to leaf through the site as they would a printed book.

Previous. Also include a Previous link returning the user to the prior page in the Web site structure, thereby reversing direction in screen navigation. The browser Back button will only return the user to the last page viewed. This will facilitate movement through a site for those entering from another Web site into the page. Leafing backward through the page hierarchy will also be easy. Locate this link at the end of the page. For long pages, also include a Previous link at the page top.

Also consider including links such as the following:

Places of related interest. Provide links to other pages with related content. Wherever the user's attention is likely to be captured, provide a direct link to related places. Also, during a search, especially when using a search facility, people rarely land directly on the desired page. Often, however, they get close. Provide links to the answers they are most likely looking for.

Important pages. Provide links to important or high-priority areas or pages you want to make sure the user sees.

Background or explanatory information. Provide links to background or explanatory information to aid users who do not have the necessary knowledge to understand or use the page. Every page must be considered independent, and its content must be understood based upon the assumption that the user has seen no other related pages. New or infrequent users may require clarification of technical concepts, a dictionary, a glossary, and an expanded discussion of important content.

Supplemental information. Use links to provide supplemental information like definitions of terms and abbreviations.

New or changed content. Draw attention to new or changed content by making it easy to notice and go directly to. A prominently placed *What's New?* link can be used for this purpose.

Quit or Exit. The Web has no way to stop running an application without closing the browser or leaving by a link. Non-Web platforms have clearly marked exit paths, including a Quit or Exit on the File Menu. Provide this command so the users can confirm that an application is finished and any entered data should be saved. This command may be included on a special exit page showing external links and other useful information.

Repeat important links. Create links to satisfy disparate user goals. Redundant links (different links to the same page) may be useful in satisfying these varied needs. Make important or critical content accessible through more than one link. Establishing multiple ways to access the same information allows users who may approach a problem from a different direction or mindset to be able to find the information.

Provide *external* links to other relevant Web sites and information sources. There is some evidence that the inclusion of outbound links increases a site's credibility. It indicates that the site authors have done their homework, and they are not afraid to let users visit other sites.

Informational sites. Links to external or foreign sites are most appropriate for informational sites, where browsing is a primary usage purpose. In applications, where a task must be completed, focusing on the task is the most important aspect of design.

Related content. Provide links to relevant information on other Web sites, including sites with similar content to that mentioned in the site being browsed. Also provide links to other resources, repositories, reference information, and background reading.

Separate page. Links that go outside of the Web site can be placed on a separate page. To accomplish this, use a *See Also* link to this additional page. Placing these links on a separate page will not disrupt the flow of the displayed pages, and will not tempt people to leave the site before they have adequately reviewed it.

Outside indication. Identify links leading away from the site by a heading or another unique designation. Also inform users that they are leaving the displayed site for another Web site (see previous discussion).

Writing Link Labels

- Create meaningful labels
 - Containing action words.
 - Positioning keywords at the beginning.
 - Long enough to be understood.
 - Clearly indicating the link destination or resulting action.
 - Matching link name with its destination page.
 - Ensure that embedded links are descriptive.
 - Provide glosses or link labels to assist link understanding.
-

Link labels must be descriptive, differentiable, and predictive. The aforementioned brief guidelines are thoroughly discussed in Step 8.

Number of Links

- Every page should contain at least one link.
 - Be conservative in the total number of links presented on a screen.
 - Restrict embedded links to those most important, pertinent, and interesting.
 - Place less relevant links in a listing.
-

One link. At a minimum, every page should contain at least one link. To follow a path and then reach a dead end is frustrating. Also, a dead-end page, if accessed from another Web site, provides no means for the user to navigate to other site pages.

Conservative number. How many links presented on a page is ultimately determined by the complexity of the site and its content. Menu research indicates that without choice groupings, links should be limited to four to eight, with groupings, limited to 18 to 24. Some experts in Web design recommend even fewer, a maximum of 8 to 12 links. In general, the more links contained on a page, the more decisions concerning which link to follow are imposed on the user. Where any link ambiguity exists, the odds of guessing correctly which link to follow diminish. A smaller number of links also increases the likelihood that those being looked for will be noticed. It is not necessary to mention all features of a site on all pages. To reduce the number of links, restrict those presented to the most important site content or useful features.

Embedded links. The trade-off that must be addressed in creating embedded links is linkability versus readability. Embedded links can be a distraction and reduce page readability, especially if used in abundance. They may also be overlooked in text scanning, especially if the scanning is not carefully done. Embedded links, however, can provide more meaningful context, adjacent phrases or sentence words being useful in understanding the link's purpose. The best trade-off is to incorporate embedded links in moderation. Reserve them for the most important, pertinent, and interesting document points. If other relevant content exists, present it in an explicit link listing.

Other Link Guidelines

In general, many of the principles in menu design presented earlier in this step, and to be presented in Step 7, also apply to presenting and organizing links. These guidelines should be reviewed in conjunction with this brief summary that follows:

- **Grouping:**
 - Group links by the most relevant menu-grouping scheme.
 - Separate visually the following types of navigation:
 - Upward to the immediate parent page.
 - Upward to the beginning of the section or category of information.
 - Across to main sections or categories of information.
 - To basic utilities.
- **Ordering:**
 - Order links by the most relevant menu choice-ordering scheme.
- **For multilevel pages, provide one simple action to**
 - Return to the next higher-level page.
 - Return to the homepage.
- **Heading:**
 - Where appropriate, provide a listing heading describing the organizing category, principle, or theme.
- **Size:**
 - Provide graphical images and command buttons of sufficient and equal size.

- **Spacing:**
 - Create equal spacing between choices graphical image and textual listing toolbars.
- **Inapplicability:**
 - Disable and display dimmed links conditionally not applicable.

Grouping. Place links of a similar purpose and function together. Develop groupings using the most relevant grouping scheme. In Web navigation, it is useful to visually separate the following types of links: (1) upward to the immediate parent page, (2) upward to the beginning of the presented section or category of information, (3) across to main sections or categories of information, and (4) to basic utilities. People make better link choices when they can readily eliminate wrong links. Grouping helps this process.

Ordering. Arrange the links by the most relevant menu-choice-ordering scheme, such as importance, frequency of use, or sequence of use, as previously described.

Multilevel pages. Navigation through menu levels should be accomplished through simple actions. It should always be very easy to return to the next higher-level page and the homepage.

Headings. When appropriate, provide an introductory word or phrase at the top of the link list as a heading. Inform viewers about the list's organizing category, principle, or theme. Establishing list context will aid users in selecting the correct link.

Size. To achieve balance, create a visually pleasing composition, make all links readily identifiable as links, create icons and command buttons of equal size. The size of any text inscribed on icons or buttons should also be consistent in size. In addition, explicit listings of textual links should be of the same size.

Spacing. To also achieve balance and a visually pleasing composition, all groups of links composed of icons, command buttons, listings of textual links should be equally spaced.

Inapplicability. Links that are irrelevant in a given situation should be disabled and displayed dimmed-out.

Link Maintenance

- Maintain correct internal links.
 - Frequently check and correct external links.
-

As sites are modified, internal links may have to be revised. Carefully check sequential pages if the Next and Previous links are used within the site. External links should also be checked and corrected frequently. Because of the volatile nature of the Web, a linked site's content may change, its location may change, or a site may cease to exist. The credibility of a site's entire content suffers if it is not properly maintained.

Maintaining a Sense of Place

As has been said several times, a sense of place — where one currently is in the labyrinth of the Web — is often difficult to maintain. A site's organizational structure is often complex, and the boundaries between sites often seem nonexistent. Navigation links can transport a person from anywhere to anywhere, as does the *Star Trek* spaceship transporter machine. (While this machine moves the human to a new environment, the Web moves the new environment to the human.) These radical shifts in context created by jumping around information space through links can be extremely confusing. It is important that one's location be continually reinforced, because people desire stability and assurance that they are where they think they are. They also need a sense of exactly where they can go from their current location.

Paper documents create a sense of where one is located through a mixture of graphical and textual cues supplied by their design, including the varying fonts and images used. Cues are also provided by the organizational scheme outlined in the table of contents, and the physical sensation of the entire document itself. Looking at where a bookmark is placed in a novel provides an excellent indication of one's location in the reading space. The answers to questions like "Can I finish before the aircraft lands and the business conference starts?" are capable of being predicted with some reliability. Electronic documents provide few of these physical cues. To provide a sense of place, plentiful and explicit cues relating to site context and organization must be provided. These cues are provided by the site's overall design characteristics and the specific orientation elements included within the Web site.

Design Characteristics That Aid in Maintaining a Sense of Place

-
- To assist maintaining a sense of place within a Web site,
 - Provide a simple hierarchical tree structure.
 - Provide ease of movement to important site features.
 - To assist maintaining a sense of place across multiple Web sites,
 - Provide consistency in all Web site design elements, including
 - Graphical identity schemes.
 - Component presentation.
 - Component organization and location.
-

Within a Web site. A simple hierarchical tree structure with obvious and linked major categories is an easily understood organization scheme. Easy identification of important site features, and ease of movement to them, is also important.

Across multiple Web sites. Design consistency contributes significantly to maintaining one's sense of place when one is moving between multiple sites. Design consistency gives a site a unique look and feel that becomes obvious as links are followed within it. Moving to a new site will be clearly evident when the design scheme changes. Consistency in the graphical identity scheme, use of colors, patterns, graphics, font styles, and so forth, will be the most noticeable aspects. Consistency in component presentation, organization, and location are also very important.

Design Elements That Aid in Maintaining a Sense of Place

- Provide a home base.
 - Use recurring navigation tools on all pages.
 - Use recurring elements on all pages.
 - Provide page numbers for sequential pages.
 - Provide ongoing feedback that shows where users are in a site.
 - Provide on-demand aids that illustrate the user's location within a site.
 - Site maps.
 - Table of contents.
 - Provide clearly written link labels.
-

Home base. As previously mentioned, a site's homepage is a stable, concrete, and safe anchor point to escape to in times of difficulty.

Recurring navigation tools. Standard navigation tools should appear on every page. In addition to creating uniformity in sight appearance, recurring tools create a more stable page environment, enhance navigation learning, and increase the user's control of the dialog.

Recurring page elements. Repeated page elements, such as titles, banners, logos, and icons, also create site uniformity. Omanson, Cline, Kilpatrick, and Dunkerton (1998) found that the page element that most significantly aided user orientation was the site logo.

Page numbers. For a long series of sequential pages, provide page numbers on each page to indicate where in the page string one is located. Another useful feature is to convert page numbers into links and present them on each page. A search, for example, may return a dozen pages of matches. At the bottom of each page inscribe, in link form, page numbers, as follows:

1 2 3 4 5 6 7 8 9 10 11 12

An estimation of document length is obtained, and the pages need not be viewed in sequential order.

Ongoing feedback showing location in Web site. Provide an historical trail, such as a breadcrumb trail, that shows where the user is located within a branch of a site. In addition to providing context for the displayed page, the trail permits easy return to any page up the trail.

On-demand aids illustrating location. Navigation aids, such as a site map or table of contents, when retrieved should show exactly where the user's current location fits within the structure of the site. The current position should be highlighted within the presented information structure. Ideally, in a site map, the complete navigation path from the homepage through intermediate pages to the current page should be presented.

Clearly written links. Labels that clearly indicate the function of the link, its destination, or its resulting action, reduce disorientation. Bad links are less likely to be followed and aimless wandering reduced.

Kinds of Graphical Menus

Providing the proper kinds of graphical menus to perform system tasks is also critical to system success. The best kind of menu to use in each situation depends on several factors. The following must be considered:

- The number of items to be presented in the menu.
- How often the menu is used.
- How often the menu contents may change.

Each kind of common graphical menu will be described in terms of purpose, advantages, disadvantages, and suggested proper usage. Design guidelines for each kind are also presented. A proper usage summary for the various kinds of menus are shown in Table 4.4 at the end of the menu discussion.

Menu Bar

- Proper usage:
 - To identify and provide access to common and frequently used application actions that take place in a wide variety of different windows.
 - A menu bar choice by itself should not initiate an action.

The highest-level graphical system menu is commonly called the menu bar. A menu bar consists of a collection of descriptions that serve as headings or titles for a series of actions on an associated pull-down menu. A menu bar choice by itself should not initiate an action.

The menu is typically arrayed in a horizontal row at the top of a window. Occasionally a menu bar is referred to as a collection of menu *titles*. In reality it is a menu in itself, and it is appropriate to simply refer to it as a menu. A menu bar is the starting point for many dialogs. Consistency in menu bar design and use will present to the user a stable, familiar, and comfortable starting point for all interactions. Menu bars are most effectively used for presenting common, frequent, or critical actions used on many windows in a variety of circumstances.

Menu bars often consist of a series of textual words, as represented in Figure 4.27. Macintosh, Presentation Manager, and Microsoft Windows illustrate examples of this textual approach. Some products have placed the choices within buttons, as represented in Figure 4.28. An example of this approach is Sun Microsystems' Open Look, which calls them *menu buttons*. There are also combinations of both. OSF/Motif presents a list of textual choices, but when one is selected, it resembles a button. Motif refers to these as *cascade buttons*.

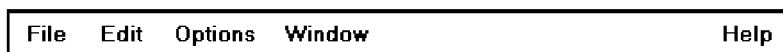


Figure 4.27: Menu bar composed of text.



Figure 4.28: Menu bar composed of buttons.

Each menu bar item is the top level of a hierarchical menu. It will have a pull-down menu associated with it, detailing the specific actions that may be performed. Some products have tried to circumvent this pull-down rule and have included items in menu bars that are direct actions themselves. These direct action items have frequently been designated by an exclamation point (!) following the menu bar description. The inclusion of direct items in a menu bar should be avoided. It creates inconsistency in menu bar use and may easily cause an action to be erroneously selected. Menu bars should always possess an associated pull-down menu.

Menu bars are used to present application alternatives or choices to the screen user. Typically, each system provides a default set of menu bar commands (for example, File, Edit, View, Window, Help).

The *advantages* of menu bars are that they

- Are always visible, reminding the user of their existence.
- Are easy to browse through.
- Are easy to locate consistently on the screen.
- Usually do not obscure the screen working area.
- Usually are not obscured by windows and dialog boxes.
- Allow for use of keyboard equivalents.

The *disadvantages* of menu bars are that

- They consume a full row of screen space.
- They require looking away from the main working area to find.
- They require moving pointer from the main working area to select.
- The menu options are smaller than full-size buttons, slowing selection time.
- Their horizontal orientation is less efficient for scanning.
- Their horizontal orientation limits number of choices that can be displayed.

Item Descriptions

- The menu item descriptions must clearly reflect the kinds of choices available.
 - Menu item descriptions will be the “titles” for pull-down menus associated with them.
 - Use mixed-case letters to describe choices.
 - Use single-word choices whenever possible.
 - Do not display choices that are never available to the user.
-

The menu item descriptions must clearly reflect alternatives available. Choices should be composed of mixed-case single words. Typically, only the first letter of the

choice is capitalized. Acronyms, abbreviations, or proper nouns that are normally capitalized may be capitalized. Choices should never be numbered.

If a multiple-word item must be used for clarity, consider including a hyphen between the multiple words to associate the words and differentiate them from other items. Do not display choices that are never available to the user.

Organization

- Follow standard platform ordering schemes where they exist.
 - Place application-specific choices where they fit best.
- Order choices left-to-right with
 - Most frequent choices to the left.
 - Related information grouped together.
- Choices found on more than one menu bar should be consistently positioned.
- Left-justify choices within the line.
- When choices can be logically grouped, provide visual logical groupings, if possible.
- Help, when included, should be located at the right side of the bar.



Figure 4.29

Follow standard platform ordering schemes where they exist. Place application-specific choices where they fit best. Order all choices left-to-right, with the most frequently elected choices to the left and related information grouped together. Choices found on more than one menu bar should be consistently positioned.

Left-justify all choices within the line (as opposed to centering them when there are not enough choices to completely fill the line). However, always locate Help, when included, at the far right side. Right side positioning will always keep Help in a consistent location within the bar. Also, provide visual groupings of all related choices, if space on the bar permits.

MAXIM Hierarchical organization is the simplest structure for visualizing and understanding complexity.

Pull-Down Menu

- Proper usage:
 - To initiate frequently used application actions that take place on a wide variety of different windows.
 - A small number of items.
 - Items best represented textually.
 - Items whose content rarely changes.
-

Selection of an alternative from the menu bar results in the display of the exact actions available to the user. These choices are displayed in a vertically arrayed listing that appears to pull down from the bar. Hence, these listings, as illustrated in Figure 4.30, are typically referred to as *pull-downs*. Other identification terms may be used, such as *drop-downs*.

Pull-downs are first-level menus used to provide access to common and frequently used application actions that take place on a wide variety of different windows. They are most useful for a small number of rarely changing items, usually about five to ten. Larger numbers of choices become awkward to use, being best handled by incorporating cascade menus (see discussion that follows). Pull-downs are best suited for items represented textually, but graphical presentations, such as colors, patterns, and shades, may also be used.

The *advantages* of pull-down menus are

- The menu bar cues a reminder of their existence.
- They may be located relatively consistently on the screen.
- No window space is consumed when they are not used.
- They are easy to browse through.
- Their vertical orientation is most efficient for scanning.
- Their vertical orientation is most efficient for grouping.
- Their vertical orientation permits more choices to be displayed.
- They allow for display of both keyboard equivalents and accelerators.

The *disadvantages* of pull-down menus are

- They require searching and selecting from another menu before seeing options.
- They require looking away from main working area to read.
- They require moving the pointer out of working area to select (unless using keyboard equivalents).
- The items are smaller than full-size buttons, slowing selection time.
- They may obscure the screen working area.

In Web use, for searching tasks, pull-down menus provide fastest performance. For browsing tasks, using the combined global/local navigation elements provided the fastest performance (Yu and Roh, 2002).



Figure 4.30: Menu bar pull-down.

Display

- Display all possible alternatives.
 - Gray-out or dim items that cannot be chosen due to the current state of an application.
-

Display all possible alternatives on a pull-down. Gray-out or dim items that cannot be chosen due to the current state of an application. If all items are, at any one point, not applicable, they must still be capable of being retrieved for perusal through the menu bar.

Size

- Must contain a minimum of two choices.
 - Restrict to no more than five to ten choices, preferably eight or less.
-

A typical pull-down consists of about five to ten choices, although more or less are sometimes seen. A pull-down should always contain more than one choice. Because of their vertical orientation, there is space for more choices containing longer descriptions than on a menu bar, and they can easily be positioned on one screen.

Organization

- Follow standard platform ordering schemes when they exist.
 - Place application-specific choices where they fit best.
 - Place frequent or critical items at the top.
 - Separate destructive choices from other choices.
 - Provide a traditional, split, or folded structure, as necessary.
 - If a folded menu is used, visually differentiate the opened choices from those high frequency choices first displayed.
 - Align choices into columns, with
 - Most frequent choices toward the top.
 - Related choices grouped together.
 - Choices found on more than one pull-down consistently positioned.
 - Left-align choice descriptions.
 - Multicolumn menus are not desirable. If necessary, organize top-to-bottom, then left-to-right.
-

Follow standard platform ordering schemes when they exist. Place application-specific choices where they fit best. Place frequent or critical items at the top of the listing, and separate destructive choices from other choices. Align all pull-down choices into

columns with their descriptions left-aligned. Locate most frequently chosen alternatives toward the top, and group related choices together. Provide a traditional, split, or folded structure, as necessary (see *Selection Support Menus* previously described). If a folded menu is used, visually differentiate the opened choices from those high frequency choices first displayed.

Choices found on more than one pull-down should be consistently positioned. Multicolumn menus are not desirable; if necessary, organize pull-downs from top-to-bottom, then left-to-right.

Groupings

- Provide groupings of related pull-down choices.
 - Incorporate a solid line between major groupings.
 - Incorporate a dotted or dashed line between subgroups.
 - Left-justify the lines under the first letter of the columnized choice descriptions.
 - Right-justify the lines under the last character of the longest choice description.
 - Display the solid line in the same color as the choice descriptions.

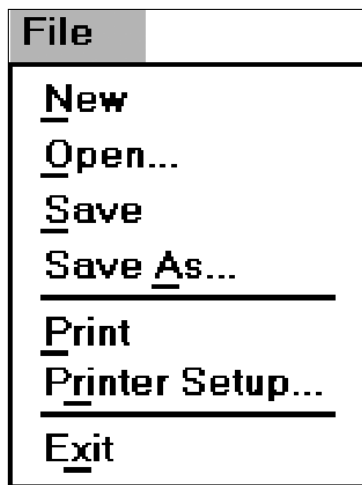


Figure 4.31

Indicate groupings of related choices by inscribing a line between each group. The line, or lines, should only extend from the first character of the descriptions to the end of the longest description, as shown above.

Some common style guides recommend that the line extend from pull-down border to border. Many other system pull-downs also follow this border-to-border approach. This extended line, however, results in too strong a visual separation between pull-down parts. The parts should be separated, but not too strongly.

Cascading Menus

-
- Proper usage:
 - To reduce the number of choices presented together for selection (reduce menu breadth).
 - When a menu specifies many alternatives and the alternatives can be grouped in meaningful related sets on a lower-level menu.
 - When a choice leads to a short, fixed list of single-choice properties.
 - When there are several fixed sets of related options.
 - To simplify a menu.
 - Avoid using for frequent, repetitive commands.
-

A *cascading* menu is a submenu derived from a higher-level menu, most typically a pull-down. Cascades may also be attached to other cascades or pop-up menus, however. Cascading menus are located to the right of the menu item on the previous menu to which they are related, as illustrated in Figure 4.32. Menu items that lead to cascading menus are typically indicated by a right-pointing triangle.

Cascading menus are developed to simplify menus by reducing the number of choices that appear together on one menu. Cascades can be used when many alternatives exist that can be grouped meaningfully. The top-level menu may contain the grouping category headings, and the cascaded menu the items in each group. Any menu choices with a fixed set of related options may utilize cascades.

The *advantages* of cascading menus are that

- The top-level menus are simplified because some choices are hidden.
- More first-letter mnemonics are available because menus possess fewer alternatives.
- High-level command browsing is easier because subtopics are hidden.

The *disadvantages* of cascading menus are

- Access to submenu items requires more steps.
- Access to submenu items requires a change in pointer movement direction.
- Exhaustive browsing is more difficult; some alternatives remain hidden as pull-downs become visible.

Changing pointer movement from a vertically oriented menu such as a pull-down to an adjacent cascade is an error-prone manual movement. Sliding the mouse and its pointer horizontally is not a very precise hand movement. As the pointer moves horizontally across the menu from which the cascade is selected it has a tendency to move vertically as well, sometimes exiting the menu over an item above or below the desired choice. When this occurs in Microsoft Windows the cascade displayed is not the one desired, but the cascade for the adjacent choice over which the pointer exited. The wrong cascade is then presented to the user, and the selection process must be repeated. Apple minimizes this problem by presenting a movement “cone” for the selected choice. This cone gradually widens as it approaches the cascade, extending somewhat over the adjacent choices. If the mouse and pointer exit the menu within an adjacent choice, but still within this cone, the originally designated cascade is still presented. The Apple solution is much more understanding of human motor limitations.

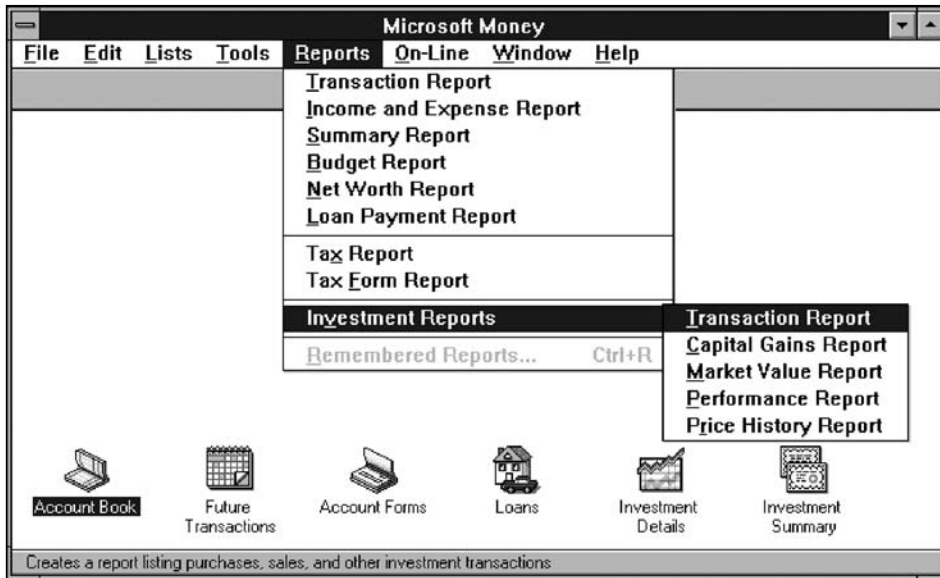


Figure 4.32: Cascading menu.

Cascade Indicator

- Place an arrow or right-pointing triangle to the right of each menu choice description leading to a cascade menu.
- Separate the indicator from the choice description by one space.
- Display the indicator in the same color as the choice descriptions.

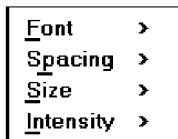


Figure 4.33

To indicate that another lower-level menu will appear when a menu item is selected, place an arrow or right-pointing triangle immediately to its right. Display the cascade indicator in the same color as the choice descriptions.

Levels

- Do not exceed three menu levels (two cascades).
 - Only one cascading menu is preferred.

Each additional cascade level presented reduces ease of access and increases visual clutter. The number of cascade levels presented should represent a balance between menu simplification, ease in menu comprehension, and ease in item selection. Whenever possible, do not exceed three levels of menus (original and two cascades). Try to limit cascades to one. If too many cascade levels are derived, create additional pull-down menus, or provide a window for some alternatives. A window is useful for establishing independent settings or the setting of multiple options. A toolbar may also be used to eliminate the necessity for traversing cascades.

Pop-Up Menus

-
- Use to present alternatives or choices within the context of the task.
-

Choices may also be presented to the user on the screen through *pop-up* menus, vertically arrayed listings that only appear when specifically requested. Pop-up menus may be requested when the mouse pointer is positioned over a designated or *hot* area of the screen (a window border or text, for example) or over a designated icon. In look, they usually resemble pull-down menus, as shown in Figure 4.34.

The kinds of choices displayed in pop-up menus are context sensitive, depending on where the pointer is positioned when the request is made. They are most useful for presenting alternatives within the context of the user's immediate task. If positioned over text, for example, a pop-up might include text-specific commands.

The *advantages* of pop-up menus are

- They appear in the working area.
- They do not use window space when not displayed.
- No pointer movement is needed if selected by button.
- Their vertical orientation is most efficient scanning.
- Their vertical orientation most efficient for grouping.
- Their vertical orientation allows more choices to be displayed.
- They may be able to remain showing ("pinned") when used frequently.
- They allow for display of both keyboard equivalents and accelerators.

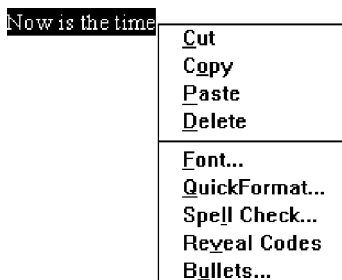


Figure 4.34: Pop-up menu.

The *disadvantages* of pop-up menus are

- Their existence must be learned and remembered.
- Means for selecting them must be learned and remembered.
- They require a special action to see the menu (mouse click).
- Items are smaller than full-size buttons, slowing selection time.
- They may obscure the screen working area.
- Their display locations may not be consistent.

For experienced users, pop-up menus are an alternative to retrieve frequently used contextual choices in pull-down menus. Choices should be limited in number and stable or infrequently changing in content.

Windows contains many contextual pop-up menus. They are also referred to as *context menus* or *shortcut menus*. Examples include the *window pop-up* and an *icon pop-up*, which presents operations of the objects represented by icons.

Display

- Provide a pop-up menu for common, frequent, contextual actions.
 - If the pointer is positioned over an object possessing more than one quality (for example, both text and graphics), at minimum present actions common to all object qualities.
 - Items that cannot be chosen due to the current state of an application should not be displayed.
 - Continue to display a pop-up until
 - A choice is selected.
 - An action outside the pop-up is initiated.
 - The user removes the pop-up.
-

Provide a pop-up menu for common, frequent, contextual actions. If the pointer is positioned over an object possessing more than one quality (for example, both text and graphics), at minimum present actions common to all object qualities. Items that cannot be chosen due to the current state of an application should not be displayed.

Continue to display a pop-up until the user selects a choice, initiates an action outside the pop-up, or requests that the pop-up be removed.

Location

- Position the pop-up
 - Centered and to the right of the object from which it was requested.
 - Close enough to the pointer so that the pointer can be easily moved onto the menu.
 - But not so close that the pointer is positioned on an item, possibly leading to accidental selection.

- If the pointer is positioned in such a manner that the pop-up would appear off-screen or clipped, position the menu
 - As close as possible to the object, but not covering the object.
 - So that it appears fully on the screen.
-

Position a pop-up menu in a consistent location relative to the object from which it is requested. The preferable location is centered to the right. Locate the pop-up close enough to the pointer so that the pointer can be easily moved onto the menu. Positioning of the pointer on the menu itself could lead to accidental selection of an action.

If the pointer is positioned in such a manner that a right-centered position would force the pop-up partially or fully off the screen, locate the pop-up fully on the screen as close as possible to the object. Do not move the pointer to make a menu fit in the most desirable location.

Size

- Restrict the pop-up to no more than five to ten choices, preferably eight or less.
-

Limit pop-up menus to about eight choices or fewer. If a large number of choices are needed, consider creating cascading menus. Minimize the number of levels of cascades, however, to provide ease of access and prevent visual clutter.

Tear-Off Menus

- Follow all relevant guidelines for pull-down menus.
-

A *tear-off* menu is a pull-down menu that can be positioned anywhere on the screen for constant referral. As such, it possesses all the characteristics of a pull-down. It may also be called a *pushpin*, *detachable*, or *roll-up* menu. Its purpose is to present alternatives or choices to the screen user that are needed infrequently at some times and heavily at other times.

Advantages/disadvantages. No space is consumed on the screen when the menu is not needed. When needed, it can remain continuously displayed. It does require extra steps to retrieve, and it may obscure the screen working area.

Tear-off menus are most useful for expert users. Use these menus in situations where the items are sometimes frequently selected and other times infrequently selected. Items should be small in number and rarely change in content. A typical use would be to detach and permanently leave displayed a pull-down menu when it must be frequently used.

Because a tear-off menu is a pull-down style, all pull-down guidelines should be followed.

Iconic Menus

- Use to remind users of the functions, commands, attributes, or application choices available.
 - Create icons that
 - Help enhance recognition and hasten option selection.
 - Are concrete and meaningful.
 - Clearly represent choices.
-

An iconic menu is the portrayal of menu items or objects in a graphic or pictorial form. The purpose of an iconic menu is to remind users of the functions, commands, attributes, or application choices available.

Advantages/disadvantages. Pictures help facilitate memory of applications, and their larger size increases speed of selection. Pictures do, however, consume considerably more screen space than text, and they are difficult to organize for scanning efficiency. To create meaningful icons requires special skills and an extended amount of time. Iconic menus should be used to designate applications or special functions within an application. Icons must be meaningful and clear. They should help enhance recognition and hasten option selection. See Step 11 for a complete review of icon design guidelines.

Pie Menus

- Consider using for
 - Mouse-driven selections, with one- or two-level hierarchies, short lists, and choices conducive to the format.
-

A *pie* menu is a circular representation of menu items, as illustrated in Figure 4.35 that can be used as an alternative to a pull-down or pop-up menu. Research has found that this style of menu yields higher performance than the typical vertical array, especially when the menu tasks are unrelated. Their basic advantage is that, when presented with the mouse pointer positioned in the pie's center, average movement to any pie wedge is shorter. Mayhew (1992) concludes that pie menus might work well for mouse-driven selections with one- or two-level hierarchies, short choice listings, and data conducive to the format. Performance advantages for keyboard selection are doubtful, however.

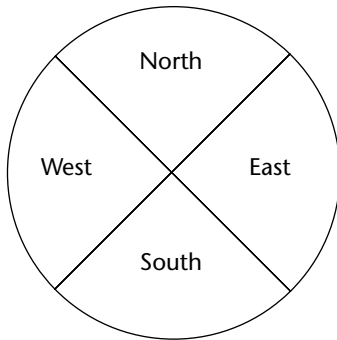


Figure 4.35: Pie menu.

Table 4.4: Menu Proper Usage Summary

Menu Bar	To identify and provide access to <ul style="list-style-type: none"> • Common and frequently used application actions. • Actions that take place in a wide variety of different windows.
Pull-Down Menu	For frequently used application actions that take place in a wide variety of different windows, <ul style="list-style-type: none"> • A small number of items (five–ten). • Items rarely changing in content.
Cascading Menu	To simplify a higher-level menu. To provide easier browsing of a higher-level menu. For mutually exclusive choices. Restrict to one–two cascades.
Pop-Up Menu	For <ul style="list-style-type: none"> • Frequent users. • Frequently used contextual commands. • A small number of items (five–ten). • Items rarely changing in content. • Items that require a small amount of screen space.
Tear-Off Menu	For items <ul style="list-style-type: none"> • Sometimes frequently selected. • Sometimes infrequently selected. • Small in number (five–ten). • Rarely changing in content.
Iconic Menu	To designate applications available. To designate special functions within an application.

Graphical Menu Examples

What follows are examples of poor and proper menu design.

Example 1

An improperly presented menu bar and pull-down.

Menu 1.1

What are the problems in the way this menu bar and pull-down menu are presented?

(1) Keyboard mnemonics are designated by capital letters. Note the uncommon shape of “foRmat,” “cuT,” and “clEAr” when the mnemonic is not the first letter of the word.

(2) Item groupings do not exist in the pull-down. The differences in basic functions are not obvious, and the more destructive operations (Undo, Clear, and Delete) are positioned close to standard actions, increasing the potential for accidental selection. (3) The keyboard accelerators are adjacent to the choice descriptions and not set off in any way. Therefore, these alternate, and supplemental, actions visually compete with choice descriptions for the viewer’s attention.

OFFICE SYSTEM					
File	Edit	foRmat	View	Options	Help
	Undo	Ctrl+Z			
	cuT	Ctrl+X			
	Copy	Ctrl+C			
	Paste	Ctrl+V			
	clEAr				
	Delete	Del			

Menu 1.1

Menu 1.2

Keyboard mnemonics are designated by underlines, not capital letters. Choice descriptions now assume more common and recognizable shapes. Groupings, through use of white space, are established for choices in the pull-down. The different functions are much more obvious and separation is provided for the destructive actions. The different groupings are visually reinforced through use of separating lines. The lines are not extended to the pull-down border so as not to completely disassociate the choices. Keyboard alternatives are right-aligned to move them further from the choice descriptions. They are also enclosed in parentheses to visually deemphasize them, thereby reducing their visual competition with the choices. Choice descriptions are now more obvious.

OFFICE SYSTEM					
<u>F</u> ile	E <u>d</u> it	<u>F</u> ormat	<u>V</u> iew	<u>O</u> ptions	<u>H</u> elp
	U <u>n</u> do	(Ctrl+Z)			
	C <u>u</u> t	(Ctrl+X)			
	<u>C</u> opy	(Ctrl+C)			
	<u>P</u> aste	(Ctrl+V)			
	C <u>l</u> ear				
	<u>D</u> elete	(Del)			

Menu 1.2

More examples and an exercise for Step 4 can be found on this book's companion Web site, www.wiley.com/college/galitz.



Future Vision

FUTURE VISION BIE

By K B Hemanth Raj

Visit : <https://hemanthrajhemu.github.io>

Quick Links for Faster Access.

CSE 8th Semester - <https://hemanthrajhemu.github.io/CSE8/>

ISE 8th Semester - <https://hemanthrajhemu.github.io/ISE8/>

ECE 8th Semester - <https://hemanthrajhemu.github.io/ECE8/>

8th Semester CSE - TEXTBOOK - NOTES - QP - SCANNER & MORE

17CS81 IOT - <https://hemanthrajhemu.github.io/CSE8/17SCHEME/17CS81/>

17CS82 BDA - <https://hemanthrajhemu.github.io/CSE8/17SCHEME/17CS82/>

17CS832 UID - <https://hemanthrajhemu.github.io/CSE8/17SCHEME/17CS832/>

17CS834 SMS - <https://hemanthrajhemu.github.io/CSE8/17SCHEME/17CS834/>

8th Semester Computer Science & Engineering (CSE)

8th Semester CSE Text Books: <https://hemanthrajhemu.github.io/CSE8/17SCHEME/Text-Book.html>

8th Semester CSE Notes: <https://hemanthrajhemu.github.io/CSE8/17SCHEME/Notes.html>

8th Semester CSE Question Paper: <https://hemanthrajhemu.github.io/CSE8/17SCHEME/Question-Paper.html>

8th Semester CSE Scanner: <https://hemanthrajhemu.github.io/CSE8/17SCHEME/Scanner.html>

8th Semester CSE Question Bank: <https://hemanthrajhemu.github.io/CSE8/17SCHEME/Question-Bank.html>

8th Semester CSE Answer Script: <https://hemanthrajhemu.github.io/CSE8/17SCHEME/Answer-Script.html>

Contribution Link:

<https://hemanthrajhemu.github.io/Contribution/>

Stay Connected... get Updated... ask your queries...

Join Telegram to get Instant Updates:

<https://telegram.me/joinchat/AAAAFTtp8kuvCHALxuMaQ>

Contact: MAIL: futurevisionbie@gmail.com

INSTAGRAM: www.instagram.com/futurevisionbie/