Atria Institute of Technology

Department of Information Science and Engineering

Bengaluru-560024

ACADEMIC YEAR: 2021-2022

ODD SEMESTER NOTES

Semester : 7th Semester

Subject Name : User Interface Design

Subject Code : 18CS734

Faculty Name : Mrs. Sonia S B

MODULE 1

CHAPTER 1. THE IMPORTANCE OF USER INTERFACE

Syllabus: The User Interface – Introduction, Overview, the importance of UI – Defining the UI, The importance of good design, Characteristics of graphical and web user interfaces, Principles of UID.

Defining the User Interface

- User interface design is a subset of a field of study called human-computer interaction (HCI). Human-computer interaction is the study, planning, and design of how people and computers work together so that a person's needs are satisfied in the most effective way.
- HCI designers must consider a variety of factors: what people want and expect, what physical limitations and abilities people possess, how their perceptual and information processing systems work, and what people find enjoyable and attractive. Technical characteristics and limitations of the computer hardware and software must also be considered.
- The user interface is the part of a computer and its software that people can see, hear, touch, talk to, or otherwise understand or direct. The user interface has essentially two components: input and output.
- Input is how a person communicates his or her needs or desires to the computer. Some common input components are the keyboard, mouse, trackball, one's finger (for touch-sensitive screens), and one's voice (for spoken instructions).
- Output is how the computer conveys the results of its computations and requirements to the user. Today, the most common computer output mechanism is the display screen, followed by mechanisms that take advantage of a person's auditory capabilities: voice and sound.
- Proper interface design will provide a mix of well-designed input and output mechanisms that satisfy the user's needs, capabilities, and limitations in the most effective way possible.
- The best interface is one which has proper design with combination of effective input and output mechanisms.

Importance of Good Design

- In spite of today's rich technologies and tools we are unable to provide effective and usable screen because lack of time and care.
- A well-designed interface and screen is terribly important to our users. It is their window to view the capabilities of the system and it is also the vehicle through which complex tasks can be performed.

- A screen's layout and appearance affect a person in a variety of ways. If they are confusing and inefficient, people will have greater difficulty in doing their jobs and will make more mistakes.
- Poor design may even chase some people away from a system permanently. It can also lead to aggravation, frustration, and increased stress.

Benefits of Good Design

- The benefits of a well-designed screen have also been under experimental scrutiny for many years. One researcher, for example, attempted to improve screen clarity and readability by making screens less crowded. The result: screen users of the modified screens completed transactions in 25 percent less time and with 25 percent fewer errors than those who used the original screens.
- Another researcher has reported that reformatting inquiry screens following good design principles reduced decision-making time by about 40 percent, resulting in a savings of 79 person-years in the affected system.
- Other benefits also accrue from good design (Karat, 1997). Training costs are lowered because training time is reduced, support line costs are lowered because fewer assist calls are necessary, and employee satisfaction is increased because aggravation and frustration are reduced.
- Another benefit is, ultimately, that an organization's customers benefit because of the improved service they receive.
- Identifying and resolving problems during the design and development process also has significant economic benefits.

CHAPTER 2

CHARACTERISTICS OF GRAPHICAL AND WEB USER INTERFACE

The Graphical User Interface

- In brief, a graphical user interface can be defined as follows. A user interface, as recently described, is a collection of techniques and mechanisms to interact with something. In a graphical interface, the primary interaction mechanism is a pointing device of some kind.
- What the user interacts with is a collection of elements referred to as objects. They can be seen, heard, touched, or otherwise perceived. Objects are always visible to the user and are used to perform tasks. They are interacted with as entities independent of all other objects.
- People perform operations, called actions, on objects. The operations include accessing and modifying objects by pointing, selecting, and manipulating.

The Popularity of Graphics

- Graphics revolutionized design and the user interface. Graphics assumes three dimensional look whereas text based system assumes one dimensional look.
- Information can appear or disappear through floating windows and navigation and commands can be done through menu or pull downs or screen controls.
- Increased computer power and the vast improvement in the display enable the user's actions to be reacted to quickly, dynamically, and meaningfully.
- If properly used graphics can reduce mental and perceptional load and increases information transfer between men and machine because of visual comparisons and simplification of the perception of structure.

The Concept of Direct Manipulation:

The term used to describe this style of interaction for graphical systems was first used by Shneiderman (1982). He called them ―direct manipulation‖ systems, suggesting that they possess the following characteristics:

- **The system is portrayed as an extension of the real world:** A person is allowed to work in a familiar environment and in a familiar way, focusing on the data, not the application and tools. The physical organization of the system, which most often is unfamiliar, is hidden from view and is not a distraction.
- **Continuous visibility of objects and actions:** objects are continuously visible. Reminders of actions to be performed are also obvious. Nelson (1980) described this concept as "virtual reality", a representation of reality that can be manipulated. Hatfield (1981) is credited with calling it ―WYSIWYG (What You See Is What You Get) and Rutkowski (1982) described it as ―transparency.

- **Actions are rapid and incremental with visible display of results:** the results of actions are immediately displayed visually on the screen in their new and current form. Auditory feedback may also be provided. The impact of a previous action is quickly seen, and the evolution of tasks is continuous and effortless.
- **Incremental actions are easily reversible:** Finally, actions, if discovered to be incorrect or not desired, can be easily undone.

Indirect Manipulation

In practice, direct manipulation of all screen objects and actions may not be feasible because of the following:

- The operation may be difficult to conceptualize in the graphical system.
- The graphics capability of the system may be limited.
- The amount of space available for placing manipulation controls in the window border may be limited.
- It may be difficult for people to learn and remember all the necessary operations and actions.

When this occurs, indirect manipulation is provided. Indirect manipulation substitutes words and text, such as pull-down or pop-up menus, for symbols, and substitutes typing for pointing. Most window systems are a combination of both direct and indirect manipulation. A menu may be accessed by pointing at a menu icon and then selecting it (direct manipulation). The menu itself, however, is a textual list of operations (indirect manipulation). When an operation is selected from the list, by pointing or typing, the system executes it as a command.

Graphical Systems: Advantages and Disadvantages

Advantages

The success of graphical systems has been attributed to a host of factors. The following have been commonly referenced in literature and endorsed by their advocates as advantages of these systems.

- **Symbols recognized faster than text:** symbols can be recognized faster and more accurately than text. An example of a good classification scheme that speeds up recognition is the icons developed for indicating the kind of message being presented to the user of the system. The text of an informational message is preceded by an “i” in a circle, a warning message by an exclamation point, and a critical message by another unique symbol. These icons allow speedy recognition of the type of message being presented.
- **Faster learning:** a graphical, pictorial representation aids learning, and symbols can also be easily learned.

- **Faster use and problem solving:** Visual or spatial representation of information has been found to be easier to retain and manipulate and leads to faster and more successful problem solving.
- **Easier remembering:** Because of greater simplicity, it is easier for casual users to retain operational concepts.
- **More natural:** In human beings, actions and visual skills emerged before languages. It has also been suggested that symbolic displays are more natural and advantageous because the human mind has a powerful image memory.
- **Exploits visual/spatial cues:** Spatial relationships are usually found to be understood more quickly than verbal representations. Visually thinking is believed to be better than logical thinking.
- **Fosters more concrete thinking:** Displayed objects are directly in the high-level task domain, or directly usable in their presented form. There is no need mentally to decompose tasks into multiple commands with complex syntactic form. The need for abstract thinking is therefore minimized.
- **Provides context:** Displayed objects are visible, providing a picture of the current context
- **Fewer errors:** Reversibility of actions reduces error rates because it is always possible to undo the last step. Error messages are less frequently needed.
- **Increased feeling of control:** The user initiates actions and feels in control. This increases user confidence and hastens system mastery.
- **Immediate feedback:** The results of actions furthering user goals can be seen immediately. If the response is not in the desired direction, the direction can be changed quickly.
- **Predictable system responses:** Predictable system responses also speed learning.
- **Easily reversible actions:** This ability to reverse unwanted actions also increases user confidence.
- **Less anxiety concerning use:** Hesitant or new users feel less anxiety when using the system because it is so easily comprehended, is easy to control, and has predictable responses and reversible actions.
- **More attractive:** Direct-manipulation systems are more entertaining, cleverer, and more appealing.
- **May consume less space:** Icons may take up less space than the equivalent in words but this is not the case always.
- **Replaces national languages:** Language translations frequently cause problems in a text-based system. Icons possess much more universality than text and are much more easily comprehended worldwide.
- **Easily augmented with text displays:** Where graphical design limitations exist, direct-manipulation systems can easily be augmented with text displays. The reverse is not true.
- **Low typing requirements:** Pointing and selection controls, such as the mouse or trackball, eliminate the need for typing skills.

- **Smooth transition from command language system:** Moving from a command language to a direct-manipulation system has been found to be easy. The reverse is not true.

Disadvantages

The body of positive research, hypotheses, and comment concerning graphical systems is being challenged by some studies, findings, and opinions that indicate that graphical representation and interaction may not necessarily always be better. Indeed, in some cases, it may be poorer than pure textual or alphanumeric displays. Sometimes arcane, and even bizarre. Among the disadvantages put forth are these:

- **Greater design complexity:** Controls and basic alternatives must be chosen from a pile of choices numbering in excess of 50. This design potential may not necessarily result in better design unless proper controls and windows are selected. Poor design can undermine acceptance.
- **Learning still necessary:** The first time one encounters many graphical systems, what to do is not immediately obvious. A severe learning and remembering requirement is imposed on many users because meanings of icons or using pointing device have to be learned.
- **Lack of experimentally-derived design guidelines:** today there is a lack of widely available experimentally-derived design guidelines. Earlier only few studies to aid in making design decisions were performed and available for today now. Consequently, there is too little understanding of how most design aspects relate to productivity and satisfaction.
- **Inconsistencies in technique and terminology:** Many differences in technique, terminology, and look and feel exist among various graphical system providers, and even among successive versions of the same system. So the user has to learn or relearn again while shifting to next terminology.
- **Working domain is the present:** While direct-manipulation systems provide context, they also require the user to work in the "present." Hulteen (1988), in a parody of "WYSIWYG," suggests "What you see is all you get." Walker (1989) argued that language takes you out of the here and now and the visually present. Language, she continues, makes it easier to find things.
- **Not always familiar:** Symbolic representations may not be as familiar as words or numbers. Numeric symbols elicit faster responses than graphic symbols in a visual search task.
- **Human comprehension limitations:** Human limitations may also exist in terms of one's ability to deal with the increased complexity of the graphical interface. The number of different icons that can be introduced is also restricted because of limitations in human comprehension. The number of different symbols a person can differentiate and deal with is much more limited than text. Recognizing icons requires much perceptual learning, abstracting ability, and intelligence. The motor skills required may

also challenge all but the most sophisticated users. Correctly double-clicking a mouse, for example, is difficult for some people

- **Window manipulation requirements:** Window handling and manipulation times are still excessive and repetitive. This wastes time an interrupts the decision making needed to perform tasks and jobs.
- **Production limitations:** The number of symbols that can be clearly produced using today's technology is still limited. A body of recognizable symbols must be produced that are equally legible and equally recognizable using differing technologies. This is extremely difficult today.
- **Few tested icons exist:** Icons must be researched, designed, tested, and then introduced into the marketplace. The consequences of poor or improper design will be confusion and lower productivity for users.
- **Inefficient for touch typists:** For an experienced touch typist, the keyboard is a very fast and powerful device. Moving a mouse or some other pointing mechanism may be slower.
- **Inefficient for expert users:** Inefficiencies develop when there are more objects and actions than can fit on the screen.
- **Not always the preferred style of interaction:** Not all users prefer a pure iconic interface. User will also prefer alternatives such as icons, icons with text or with textual captions.
- **Not always fastest style of interaction:** graphic instructions on an automated bank teller machine were inferior to textual instructions.
- **Increased chances of clutter and confusion:** A graphical system does not guarantee elimination of clutter on a screen. Instead, the chance for clutter is increased, thereby increasing the possibility of confusion. When working with a window, expand it to fill the entire display screen. This may be done to reduce visual screen clutter. Mori and Hayashi (1993) found that visible windows, not the focus of attention, degraded performance in the window being worked on.
- **The futz and fiddle factor:** With the proliferation of computer games, computer usage can be wasteful of time. Stromoski (1993) estimates that five hours a week in the office are spent playing and tinkering. Experts have said that the most used program in Microsoft Windows is Solitaire! Tinkering includes activities such as creating garish documents reflecting almost every object property (font size, style, colour, and so on.) available. Futzing and fiddling does have some benefits, however. It is a tool for learning how to use a mouse, for example, and it is a vehicle for exploring the system and becoming familiar with its capabilities. It is of value when done in moderation.
- **May consume more screen space:** Not all applications will consume less screen space. A listing of names and telephone numbers in a textual format will be more efficient to scan than a card file.
- **Hardware limitations:** Good design also requires hardware of adequate power, processing speed, screen resolution, and graphic capability. Insufficiencies in these areas can prevent a graphic system's full potential from being realized.

Characteristics of the Graphical User Interface

A graphical system possesses a set of defining concepts.

- **Sophisticated Visual Presentation:** Visual presentation is the visual aspect of the interface. It is what people see on the screen. The sophistication of a graphical system permits displaying lines, including drawings and icons. It also permits the displaying of a variety of character fonts, including different sizes and styles. The meaningful interface elements visually presented to the user in a graphical system include windows (primary, secondary, or dialog boxes), menus (menu bar, pull down, pop-up, cascading), icons to represent objects such as programs or files, assorted screen-based controls (text boxes, list boxes, combination boxes, settings, scroll bars, and buttons), and a mouse pointer and cursor. The objective is to reflect visually on the screen the real world of the user as realistically, meaningfully, simply, and clearly as possible.
- **Pick-and-Click Interaction:** Elements of a graphical screen upon which some action is to be performed must first be identified. The motor activity required of a person to identify a proposed action is commonly referred to as pick, the signal to perform an action as click. The primary mechanism for performing this pick-and-click is most often the mouse and its buttons and the secondary mechanism for performing these selection actions is the keyboard.
- **Restricted Set of Interface Options:** The array of alternatives available to the user is what is presented on the screen or what may be retrieved through what is presented on the screen, nothing less, and nothing more. This concept fostered the acronym WYSIWYG.
- **Visualization:** Visualization is a cognitive process that allows people to understand information that is difficult to perceive, because it is either too voluminous or too abstract. The goal is not necessarily to reproduce a realistic graphical image, but to produce one that conveys the most relevant information. Effective visualizations can facilitate mental insights, increase productivity, and foster faster and more accurate use of data.
- **Object Orientation:** A graphical system consists of objects and actions. Objects are what people see on the screen as a single unit. Objects can be composed of subobjects. For example, an object may be a document and its subobjects may be a paragraph, sentence, word, and letter.
  Objects are divided into three meaningful classes
  1. Data objects, which present information either text or graphics in the body of the screen
  2. Container objects to hold other objects. They are used to group two or more related objects for easy access and retrieval. There are three kinds of container objects: the workplace, folders, and workareas. The workplace is the desktop, the storage area for all objects. Folders are general-purpose containers for long-term storage of objects. Workareas are temporary storage folders used for storing multiple objects currently being worked on.

3. Device objects represent physical objects in the real world, such as printers or trash baskets. These objects may contain others for acting upon. A file, for example, may be placed in a printer for printing of its contents. Microsoft Windows specifies the characteristics of objects depending upon the relationships that exist between them.

Objects can exist within the context of other objects, and one object may affect the way another object appears or behaves. These relationships are called collections, constraints, composites, and containers.

A collection is the simplest relationship—the objects sharing a common aspect. A collection might be the result of a query or a multiple selection of objects. Operations can be applied to a collection of objects.

A constraint is a stronger object relationship. Changing an object in a set affects some other object in the set. A document being organized into pages is an example of a constraint.

A composite exists when the relationship between objects becomes so significant that the aggregation itself can be identified as an object. Examples include a range of cells organized into a spreadsheet, or a collection of words organized into a paragraph.

A container is an object in which other objects exist. Examples include text in a document or documents in a folder. A container often influences the behavior of its content. It may add or suppress certain properties or operations of objects placed within it, control access to its content, or control access to kinds of objects it will accept. These relationships help define an object's type. Similar traits and behaviors exist in objects of the same object type.

Another important object characteristic is persistence. Persistence is the maintenance of a state once it is established. An object's state (for example, window size, cursor location, scroll position, and so on) should always be automatically preserved when the user changes it.

- **Properties or Attributes of Objects:** Properties are the unique characteristics of an object. Properties help to describe an object and can be changed by users.
  Actions: People take actions on objects. They manipulate objects in specific ways (commands) or modify the properties of objects (property or attribute specification). Commands are actions that manipulate objects. They may be performed in a variety of ways, including by direct manipulation or through a command button. Examples of commands are opening a document, printing a document, closing a window, and quitting an application. Property/attribute specification actions establish or modify the attributes or properties of objects. When selected, they remain in effect until deselected.

Examples include selecting cascaded windows to be displayed, a particular font style, or a particular colour.
The following is a typical property/attribute specification sequence:
  - The user selects an object—for example, several words of text.
  - The user then selects an action to apply to that object, such as the action BOLD.
  - The selected words are made bold and will remain bold until selected and changed again.

- **Application versus Object or Data Orientation:** Earlier graphical systems were usually application-oriented, a continuation of the philosophy that enveloped text-based systems. When a text-based system was developed, it was called an application. As graphical systems evolved, developers usually thought in terms of applications as well. When a real picture of the user began to emerge, it finally became evident that people thought in terms of tasks, not applications. They choose objects and then act upon them.
  An application-oriented approach takes an action: object approach, like this:
  Action> 1. An application is opened (for example, word processing).
  Object> 2. A file or other object selected (for example, a memo).
  An object-oriented object: action approach does this:
  Object> 1. An object is chosen (a memo).
  Action> 2. An application is selected (word processing).
  The object-action approach permits people to more easily focus on their task and minimizes the visibility of the operating system and separate applications.
- **Views:** Views are ways of looking at an object's information. IBM's SAA CUA describes four kinds of views: composed, contents, settings, and help. Composed views present information and the objects contained within an object. They are typically associated with data objects and are specific to tasks and products being worked with. Contents views list the components of objects. Settings views permit seeing and changing object properties. Help views provide all the help functions.
- **Use of Recognition Memory:** Continuous visibility of objects and actions encourages to eliminate "out of sight, out of mind" problem.
- **Concurrent Performance of Functions:** Graphic systems may do two or more things at one time. Multiple programs may run simultaneously. It may process background tasks (cooperative multitasking). When applications are running as truly separate tasks, the system may divide the processing power into time slices and allocate portions to each application (preemptive multitasking). Data may also be transferred between programs. It may be temporarily stored on a ―clipboard‖ for later transfer or be automatically swapped between programs.

The Web User Interface

- Web interface design is essentially the design of navigation and the presentation of information.
- Proper interface design is largely a matter of properly balancing the structure and relationships of menus, content, and other linked documents or graphics. The design

goal is to build a hierarchy of menus and pages that feels natural, is well structured, is easy to use, and is truthful.

- The Web is a navigation environment where people move between pages of information, not an application environment. It is also a graphically rich environment.
- Web interface design is difficult for a number of reasons. First, its underlying design language, HTML. Next, browser navigation retreated to the pre-GUI era.
- Web interface design is also more difficult because the main issues concern information architecture and task flow, neither of which is easy to standardize. It is more difficult because of the availability of the various types of multimedia, and the desire of many designers to use something simply because it is available. It is more difficult because users are ill defined, and the user's tools so variable in nature.

The Popularity of the Web

- While the introduction of the graphical user interface revolutionized the user interface, the Web has revolutionized computing. It allows millions of people scattered across the globe to communicate, access information, publish, and be heard. It allows people to control much of the display and the rendering of Web pages.
- Web usage has reflected this popularity. The number of Internet hosts has risen dramatically.
- Users have become much more discerning about good design. Slow download times, confusing navigation, confusing page organization, disturbing animation, or other undesirable site features often results in user abandonment of the site for other with a more agreeable interface.

Characteristics of a Web Interface

A Web interface possesses a number of characteristics, some similar to a GUI interface, and, as has already been shown, some different.

GUI versus Web Page Design

GUI and Web interface design do have similarities. Both are software designs, they are used by people, they are interactive, they are heavily visual experiences presented through screens, and they are composed of many similar components.

Table 2.1 GUI versus Web Design

	GUI	WEB
Devices	User hardware variations limited. User hardware characteristics well defined. Screens appear exactly as specified.	User hardware variations enormous. Screen appearance influenced by hardware being used.
User Focus	Data and applications.	Information and navigation.
Data/ Information	Typically created and used by known and trusted sources. Properties generally known. Typically placed into system by users or known people and organizations. Typically organized in a meaningful fashion. A notion of private and shared data exists.	Full of unknown content. Source not always trusted. Often not placed onto the Web by users or known people and organizations. Highly variable organization. Privacy often suspect.
User Tasks	Install, configure, personalize, start, use, and upgrade programs. Open, use, and close data files. Fairly long times spent within an application. Familiarity with applications often achieved.	Link to a site, browse or read pages, fill out forms, register for services, participate in transactions, download and save things. Movement between pages and sites very rapid. Familiarity with many sites not established.
User's Conceptual Space	Controlled and constrained by program.	Infinite and generally unorganized.
Presentation Elements	Windows, menus, controls, data, toolbars, messages, and so on. Many transient, dynamically appearing and disappearing. Presented as specified by designer. Generally standardized by toolkits and style guides.	Two components, browser and page. Within page, any combination of text, images, audio, video, and animation. May not be presented as specified by the designer–dependent on browser, monitor, and user specifications. Little standardization.

Navigation	Through menus, lists, trees, dialogs, and wizards. Not a strong and visible concept. Constrained by design. Generally standardized by toolkits and style guides.	Through links, bookmarks, and typed URLs. Significant and highly visible concept. Few constraints, frequently causing a lost "sense of place." Few standards. Typically part of page design, fostering a lack of consistency.
Context	Enables maintenance of a better sense of context. Restricted navigation paths. Multiple viewable windows.	Poorer maintenance of a sense of context. Single-page entities. Unlimited navigation paths. Contextual clues become limited or are difficult to find.
Interaction	Interactions such as clicking menu choices, pressing buttons, selecting list choices, and cut/copy/paste occur within context of active program.	Basic interaction is a single click. This can cause extreme changes in context, which may not be noticed.
Response Time	Nearly instantaneous.	Quite variable, depending on transmission speeds, page content, and so on. Long times can upset the user.
Visual Style	Typically prescribed and constrained by toolkit. Visual creativity allowed but difficult. Little significant personalization.	Fosters a more artistic, individual, and unrestricted presentation style. Complicated by differing browser and display capabilities, and bandwidth limitations. Limited personalization available.
System Capability	Unlimited capability proportional to sophistication of hardware and software.	Limited by constraints imposed by the hardware, browser, software, client support, and user willingness to allow features because of response time, security, and privacy concerns.
Task Efficiency	Targeted to a specific audience with specific tasks. Only limited by the amount of programming undertaken to support it.	Limited by browser and network capabilities. Actual user audience usually not well understood. Often intended for anyone and everyone.

In conclusion, from a design implication perspective, GUI and Web differences can be extensive. Today, these differences must be considered in any Web page design, although many GUI interface design techniques and guidelines are applicable in page design. Tomorrow, many of these GUI-Web differences will diminish or disappear as the discrepancies are addressed by technology.

Table 2.1 *(Continued)*

	GUI	WEB
Consistency	Major objective exists within and across applications. Aided by platform toolkit and design guidelines. Universal consistency in GUI products generally created through toolkits and design guidelines.	Sites tend to establish their own identity. Frequently standards set within a site. Frequent ignoring of GUI guidelines for identical components, especially controls.
User Assistance	Integral part of most systems and applications. Accessed through standard mechanisms. Documentation, both online and offline, usually provided. Personal support desk also usually provided.	No similar help systems. The little available help is built into the page. Customer service support, if provided, oriented to product or service offered.
Integration	Seamless integration of all applications into the platform environment a major objective. Toolkits and components are key elements in accomplishing this objective.	Apparent for some basic functions within most Web sites (navigation, printing, and so on.) Sites tend to achieve individual distinction rather than integration.
Security	Tightly controlled, proportional to degree of willingness to invest resources and effort. Not an issue for most home PC users.	Renowned for security exposures. Browser-provided security options typically not understood by average users. When employed, may have function-limiting side effects.
Reliability	Tightly controlled in business systems, proportional to degree of willingness to invest resources and effort.	Susceptible to disruptions caused by user, telephone line and cable providers, Internet service providers, hosting servers, and remotely accessed sites.

Printed Pages versus Web Pages

- **Page size:** Printed pages are generally larger than their Web counterparts. They are also fixed in size, not variable like Web pages. The visual impact of the printed page is maintained in hard-copy form, while on the Web all that usually exists are snapshots of page areas. The visual impact of a Web page is substantially degraded, and the user may never see some parts of the page because their existence is not known or require scrolling to bring into view. The design implications: the top of a Web page is its most important element, and signals to the user must always be provided that parts of a page lie below the surface.
- **Page rendering:** Printed pages are immensely superior to Web pages in rendering. Printed pages are presented as complete entities, and their entire contents are available for reading or review immediately upon appearance. Web pages elements are often rendered slowly, depending upon things like line transmission speeds and page content. Design implications: Provide page content that downloads fast, and give people elements to read immediately so the sense of passing time is diminished.

- **Page layout:** With the printed page, layout is precise with much attention given to it. With Web pages layout is more of an approximation, being negatively influenced by deficiencies in design toolkits and the characteristics of the user's browser and hardware, particularly screen sizes. Design implication: Understand the restrictions and design for the most common user tools.
- **Page resolution:** the resolution of displayed print characters still exceeds that of screen characters, and screen reading is still slower than reading from a document. Design implication: Provide an easy way to print long Web documents.
- **User focus:** Printed pages present people with entire sets of information. Estimations of effort needed to deal with the document are fairly easily made, size and the nature of the material being strong contributors. Some printed pages may be read sequentially (a novel) and others (a newspaper) partially and somewhat sequentially (the sports section first, perhaps?). Others forms of printed material may simply be systematically skimmed (a sales brochure). Web pages present people with individual snapshots of information, often with few clues as to structure and sequence, and rarely with a few cues as to length and depth. Web information potentially available is almost unlimited, and that information paths can lead everywhere and anywhere. With few content size cues available and a huge information base, a common resulting behavior of Web users is to skim the information presented, looking for what is most relevant to their task or need. This is done for personal efficiency and so as not to tax one's patience. Design implications: Create easy to scan pages and limit the word count of textual content. Also, provide overviews of information organization schemes, clear descriptions of where links lead, and estimations of sizes of linked pages and materials.
- **Page navigation:** Navigating printed materials is as simple as page turning. Navigating the Web requires innumerable decisions concerning which of many possible links should be followed. Design implications are similar to the above provide overviews of information organization schemes and clear descriptions of where links lead. • Interactivity: Printed page design involves letting the eyes traverse static information, selectively looking at information and using spatial combinations to make page elements enhance and explain each other. Web design involves letting the hands move the information (scrolling, pointing, expanding, clicking, and so on) in conjunction with the eyes.
- **Sense of place:** The document is an object with physical characteristics. Paging through printed material is an orderly process, sequential and understandable. Electronic documents provide none of these physical cues. All that is visible is a small collection of text, graphics, and links hinting that much else lies somewhere underneath. Moving through the Web links can cause radical shifts in location and context. Paging using the browser's Back button steps one back through links visited and may involve passing through different documents. All these factors cause a person to easily lose a sense of place and lead to confusion. Design implication: Build cues into Web pages to aid the user in maintaining a sense of place
- **Interactivity:** Printed page design involves letting the eyes traverse static information, selectively looking at information and using spatial combinations to make page

elements enhance and explain each other. Web design involves letting the hands move the information (scrolling, pointing, expanding, clicking, and so on) in conjunction with the eyes. Information relationships, static or dynamic, are expressed chronologically as part of the interaction and user movements. Doing is more memorable and makes a stronger impact than simply seeing. No print precedents exist for this style of interaction. A better understanding of this process (as well as better hardware and software) is needed to enhance interactivity

- **Page independence:** Because moving between Web pages is so easy, and almost any page in a site can be accessed from anywhere else, pages must be made freestanding. Every page is independent. Printed pages, being sequential, fairly standardized in organization, and providing a clear sense of place, are not considered independent. Design implication: Provide informative headers and footers on each Web page.

In conclusion, many of the basic print guidelines can be applied to Web page design. As shown above, however, printed material design differs from Web page design in many aspects. New guidelines must continue to be developed, implemented, and modified as necessary as technology advances and our understanding of Web interaction increases. For the moment, apply existing guidelines where relevant, and new guidelines as necessary

The Merging of Graphical Business Systems and the Web

Strength of the Web lies in its ability to link databases and processing occurring on a variety of machines within a company or organization. The graphical business system and the Web will merge into a common entity. These Web systems are called intranets.

Characteristics of an Intranet versus the Internet

They differ, however, in some important ways as

- **Users:** The users of intranets, being organization employees, know a lot about the organization, its structure, its products, its jargon, and its culture. Customers use Internet sites and others who know much less about the organization, and often care less about it. The intranet user's characteristics and needs can be much more specifically defined then can those of the general Internet use.
- **Tasks:** An intranet is used for an organization's everyday activities, including complex transactions, queries, and communications. The Internet is mainly used to find information, with a supplementary use being simple transactions.
- **Type of information:** An intranet will contain detailed information needed for organizational functioning. Information will often be added or modified. The Internet will usually present more stable information: marketing and customer or client information, reports, and so forth.
- **Amount of information:** Typically, an intranet site will be much larger than an organization's Internet site. It has been estimated that an intranet site can be ten to one hundred times larger than its corresponding public site.

- **Hardware and software:** Since intranets exist in a controlled environment, the kinds of computers, monitors, browsers, and other software can be restricted or standardized. The need for cross-platform compatibility is minimized or eliminated; upgraded communications also permit intranets to run from a hundred to a thousand times faster than typical Internet access can. This allows the use of rich graphics and multimedia, screen elements that contribute to very slow download times for most Internet users.
- **Design philosophy:** Implementation on the intranet of current text-based and GUI applications will present a user model similar to those that have existed in other domains. This will cause a swing back to more traditional GUI designs—designs that will also incorporate the visual appeal of the Web, but eliminate many of its useless, promotional, and distracting features. The resulting GUI hybrids will be richer and much more effective.

Extranets

- An extranet is a special set of intranet Web pages that can be accessed from outside an organization or company.
- Typical examples include those for letting customers check on an order's status or letting suppliers view requests for proposals. An extranet is a blend of the public Internet and the intranet, and its design should reflect this.

Principles of User Interface Design

An interface must really be just an extension of a person. This means that the system and its software must reflect a person's capabilities and respond to his or her specific needs. It should be useful, accomplishing some business objectives faster and more efficiently than the previously used method or tool did. It must also be easy to learn, for people want to do, not learn to do. The interface itself should serve as both a connector and a separator: a connector in that it ties the user to the power of the computer, and a separator in that it minimizes the possibility of the participants damaging one another. We will begin with the first set of published principles, those for the Xerox STAR.

Principles for the Xerox STAR

These principles established the foundation for graphical interfaces.

- **The illusion of manipulable objects**: Displayed objects that are selectable and manipulable must be created. A design challenge is to invent a set of displayable objects that are represented meaningfully and appropriately for the intended application. It must be clear that these objects can be selected. The handles for windows were placed in the borders (window specific commands, pop-up menus, scroll bars, and so on).
- **Visual order and viewer focus:** Effective visual contrast between various components of the screen is used to achieve this goal. Animation is also used to draw attention, as is sound. Feedback must also be provided to the user.

- **Revealed structure:** The distance between one's intention and the effect must be minimized. The relationship between intention and effect must be tightened and made as apparent as possible to the user.
- **Consistency:** Consistency aids learning. Consistency is provided in such areas as element location, grammar, font shapes, styles, and sizes, selection indicators, and contrast and emphasis techniques.
- **Appropriate effect or emotional impact:** The interface must provide the appropriate emotional effect for the product and its market. Is it a corporate, professional, and secure business system? Should it reflect the fantasy, wizardry, and bad puns of computer games?
- **A match with the medium:** The interface must also reflect the capabilities of the device on which it will be displayed. Quality of screen images will be greatly affected by a device's resolution and color-generation capabilities.

General Principles

The design goals in creating a user interface are described below. They are fundamental to the design and implementation of all effective interfaces, including GUI and Web ones. These principles are general characteristics of the interface, and they apply to all aspects.

- **Aesthetically Pleasing**: Provide visual appeal by following these presentation and graphic design principles:
  - Provide meaningful contrast between screen elements.
  - Create groupings.
  - Align screen elements and groups.
  - Provide three-dimensional representation.
  - Use colour and graphics effectively and simply.

- **Clarity:** The interface should be visually, conceptually, and linguistically clear, including:
  - Visual elements
  - Functions
  - Metaphors
  - Words and text

- **Compatibility:**
  - ❖ Provide compatibility with the following:
    - The user
    - The task and job
    - The product
  - ❖ Adopt the user's perspective.

- **Comprehensibility:**

- ❖ A system should be easily learned and understood. A user should know the following:
  - What to look at
  - What to do
  - When to do it
  - Where to do it
  - Why to do it
  - How to do it
- ❖ The flow of actions, responses, visual presentations, and information should be in a sensible order that is easy to recollect and place in context.

- **Configurability:** — Permit easy personalization, configuration, and reconfiguration of settings.
  - Enhances a sense of control.
  - Encourages an active role in understanding.

- **Consistency:**
  - ❖ A system should look, act, and operate the same throughout. Similar components should:
    - Have a similar look.
    - Have similar uses.
    - Operate similarly.
  - ❖ The same action should always yield the same result.
  - ❖ The function of elements should not change.
  - ❖ The position of standard elements should not change.

In addition to increased learning requirements, inconsistency in design has a number of other prerequisites and by-products, including:

- More specialization by system users.
- Greater demand for higher skills.
- More preparation time and less production time.
- More frequent changes in procedures.
- More error-tolerant systems (because errors are more likely).
- More kinds of documentation.
- More time to find information in documents.
- More unlearning and learning when systems are changed.
- More demands on supervisors and managers.
- More things to do wrong.

- **Control:**
  - ❖ The user must control the interaction.
    - Actions should result from explicit user requests.
    - Actions should be performed quickly.

- Actions should be capable of interruption or termination.
- The user should never be interrupted for errors.

❖ The context maintained must be from the perspective of the user.
❖ The means to achieve goals should be flexible and compatible with the user's skills, experiences, habits, and preferences.
❖ Avoid modes since they constrain the actions available to the user.
❖ Permit the user to customize aspects of the interface, while always providing a proper set of defaults.

- **Directness:**
  ❖ Provide direct ways to accomplish tasks.
    - Available alternatives should be visible.
    - The effect of actions on objects should be visible.

- **Efficiency:**
  ❖ Minimize eye and hand movements, and other control actions.
    - Transitions between various system controls should flow easily and freely.
    - Navigation paths should be as short as possible.
    - Eye movement through a screen should be obvious and sequential.
  ❖ Anticipate the user's wants and needs whenever possible.

- **Familiarity:**
  ❖ Employ familiar concepts and use a language that is familiar to the user.
  ❖ Keep the interface natural, mimicking the user's behaviour patterns.
  ❖ Use real-world metaphors.

- **Flexibility:**
  ❖ A system must be sensitive to the differing needs of its users, enabling a level and type of performance based upon:
    - Each user's knowledge and skills.
    - Each user's experience.
    - Each user's personal preference.
    - Each user's habits.
    - The conditions at that moment.

- **Forgiveness:**
  ❖ Tolerate and forgive common and unavoidable human errors.
  ❖ Prevent errors from occurring whenever possible.
  ❖ Protect against possible catastrophic errors.
  ❖ When an error does occur, provide constructive messages.

- **Predictability:**

- ❖ The user should be able to anticipate the natural progression of each task.
  - Provide distinct and recognizable screen elements.
  - Provide cues to the result of an action to be performed.
- ❖ All expectations should be fulfilled uniformly and completely.

- **Recovery:**
  - ❖ A system should permit:
    - Commands or actions to be abolished or reversed.
    - Immediate return to a certain point if difficulties arise.
  - ❖ Ensure that users never lose their work as a result of:
    - An error on their part.
    - Hardware, software, or communication problems.

- **Responsiveness:**
  - ❖ The system must rapidly respond to the user's requests.
  - ❖ Provide immediate acknowledgment for all user actions:
    - Visual.
    - Textual.
    - Auditory.

- **Simplicity:**
  - ❖ Provide as simple an interface as possible.
  - ❖ Five ways to provide simplicity:
    - Use progressive disclosure, hiding things until they are needed.
    - Present common and necessary functions first.
    - Prominently feature important functions.
    - Hide more sophisticated and less frequently used functions.
    - Provide defaults.
    - Minimize screen alignment points.
    - Make common actions simple at the expense of uncommon actions being made harder.
    - Provide uniformity and consistency.

- **Transparency:**
  - ❖ Permit the user to focus on the task or job, without concern for the mechanics of the interface.
    ♣ Workings and reminders of workings inside the computer should be invisible to the user.

- **Trade-Offs:**
  - ❖ Final design will be based on a series of trade-offs balancing often-conflicting design principles
  - ❖ People's requirements always take precedence over technical requirements.

MODULE 2

CHAPTER 3. THE USER INTERFACE DESIGN PROCESS

Syllabus: The User Interface Design process- Obstacles, Usability, Human characteristics in design, Human Interaction speeds, Business functions- Business definition and requirement analysis, Basic business functions, Design standards.

Obstacles and Pitfalls in the Development Path

Developing a computer system is never easy. The path is littered with obstacles and traps, many of them human in nature.

Gould (1988) has made these general observations about design:

- Nobody ever gets it right the first time.
- Development is chock-full of surprises.
- Good design requires living in a sea of changes.
- Making contracts to ignore change will never eliminate the need for change.
- Even if you have made the best system humanly possible, people will still make mistakes when using it.
- Designers need good tools.
- You must have behavioural design goals like performance design goals.

The first five conditions listed will occur naturally because people are people, both as users and as developers. These kinds of behaviour must be understood and accepted in design. User mistakes, while they will always occur, can be reduced.

Pitfalls in the design process exist because of a flawed design process, including a failure to address critical design issues, an improper focus of attention, or development team organization failures. Common pitfalls are:

- No early analysis and understanding of the user's needs and expectations.
- A focus on using design features or components that are ―neat or ―glitzy.
- Little or no creation of design element prototypes.
- No usability testing.
- No common design team vision of user interface design goals.
- Poor communication between members of the development team.

Designing for People: The Five Commandments

The complexity of a graphical or Web interface will always magnify any problems that do occur. Pitfalls can be eliminated if the following design commandments remain foremost in the designer's mind.

1. **Gain a complete understanding of users and their tasks:** The users are the customers. Today, people expect a level of design sophistication from all interfaces, including Web sites. The product, system or Web site must be geared to people's needs, not those of the developers.
2. **Solicit early and ongoing user involvement:** Involving the users in design from the beginning provides a direct conduit to the knowledge they possess about jobs, tasks, and needs. Involvement also allows the developer to confront a person's resistance to change, a common human trait. People dislike change for a variety of reasons, among them fear of the unknown and lack of identification with the system.
3. **Perform rapid prototyping and testing:** Prototyping and testing the product will quickly identify problems and allow you to develop solutions. Prototyping and testing must be continually performed during all stages of development to uncover all potential defects. If thorough testing is not performed before product release, the testing will occur in the user's office. Encountering a series of problems early in system use will create a negative first impression in the customer's mind, and this may harden quickly, creating attitudes that may be difficult to change. It is also much harder and more costly to fix a product after its release.
4. **Modify and iterate the design as much as necessary:** While design will proceed through a series of stages, problems detected in one stage may force the developer to revisit a previous stage.. Establish user performance and acceptance criteria and continue testing and modifying until all design goals are met.
5. **Integrate the design of all the system components:** The software, the documentation, the help function, and training needs are all important elements of a graphical system or Web site and all should be developed concurrently. Time will also exist for design trade-offs to be thought out more carefully.

Usability

The term usability used to describe the effectiveness of human performance. The term usability is defined as ―the capability to be used by humans easily and effectively, where, easily = to a specified level of subjective assessment, effectively = to a specified level of human performance.

Common Usability Problems

Mandel (1994) lists the 10 most common usability problems in graphical systems as reported by IBM usability specialists. They are:

1. Ambiguous menus and icons.
2. Languages that permit only single-direction movement through a system.
3. Input and direct manipulation limits.
4. Highlighting and selection limitations.

5. Unclear step sequences.
6. More steps to manage the interface than to perform tasks.
7. Complex linkage between and within applications.
8. Inadequate feedback and confirmation.
9. Lack of system anticipation and intelligence.
10. Inadequate error messages, help, tutorials, and documentation.

The Web, with its dynamic capabilities and explosive entrance into our lives, has unleashed what seems like more than its own share of usability problems. Many are similar to those outlined above. Web usability characteristics particularly wasteful of people's time, and often quite irritating, are:

Visual clutter: A lack of "white space," meaningless graphics, and unnecessary and wasteful decoration often turn pages into jungles of visual noise. Meaningful content lies hidden within the unending forest of vines and trees, forcing the user to waste countless minutes searching for what is relevant. Useless displayed elements are actually a form of visual noise.

Impaired information readability: Page readability is diminished by poor developer choices in typefaces, colours, and graphics. A person's attention is directed towards trying to understand why the differences exist, instead of being focused toward identifying and understanding the page's content. Backgrounds that are brightly coloured or contain pictures or patterns greatly diminish the legibility of the overwritten text.

Incomprehensible components: Some design elements give the user no clue as to their function. Some icons and graphics, containing no text to explain what they do. Some buttons don't look at all like command buttons. Command buttons or areas that give no visual indication that they are clickable often won't be clicked. Language is also often confusing, with the developer's terminology being used, not that of the user.

Annoying distractions: Elements constantly in motion, scrolling marquees or text, blinking text, or looping continually running animations compete with meaningful content for the user's eye's and attention—and destroy a page's readability. Automatically presented music or other sounds interrupt one's concentration, as do non-requested pop-up widows, which must be removed, wasting more of the user's time. A person's senses are under constant attack, and the benefits afforded by one's peripheral vision are negated.

Confusing navigation: Poor, little, or no organization exists among pages. The size and depth of many Web sites can eventually lead to a "lost in space". Navigation links lead to dead-ends from which there is no return, or boomerang you right back to the spot where you are standing without you being aware of it. Some navigation elements are invisible. (See mystery icons and minesweeping above.) Confusing navigation violates expectations and results in disturbing unexpected behaviour.

Inefficient navigation: A person must transverse content-free pages to find what is meaningful. One whole screen is used to point to another. Large graphics waste screen space and add to the page count. The path through the navigation maze is often long and tedious.

Inefficient operations: Time is wasted doing many things. Page download times can be excessive. Pages that contain, for example, large graphics and maps, large chunky headings, or many colours, take longer to download than text. Excessive information fragmentation can require navigation of long chains of links to reach relevant material.

Excessive or inefficient page scrolling: Long pages requiring scrolling frequently lead to the user's losing context as related information's spatial proximity increases and some information entirely disappears from view and, therefore, from memory. Out of sight is often out of mind.

Information overload: Poorly organized or large amounts of information taxes one's memory and can be overwhelming. Heavy mental loads can result from making decisions concerning which links to follow and which to abandon, given the large number of choices available. Or from trying to determine what information is important, and what is not. Or from trying to maintain ones place in a huge forest of information trees.

Design inconsistency: It is expected that site differences will and must exist because each Web site owner strives for its own identity. For the user's sake, however, some consistency must exist to permit a seamless flow between sites. For example, navigation element location on a page and the look of navigation buttons (raised).

Outdated information: One important value of a Web site is its "currentness." Outdated information destroys a site's credibility in the minds of many users, and therefore its usefulness. A useless site is not very usable.

Stale design caused by emulation of printed documents and past systems: The Web is a new medium with expanded user interaction and information display possibilities. While much of what we have learned in the print world and past information systems interface design can be ported to the Web, all of what we know should not be blindly moved from one to the other. Web sites should be rethought and redesigned using the most appropriate and robust design techniques available.

Some of these usability problems are a result of the Web's "growing pains." For other problems developers themselves can only be blamed, for they too often have created a product to please themselves and "look cool," not to please their users. Symptoms of this approach include overuse of bleeding edge technology, a focus on sparkle, and jumping to implement the latest Internet technique or buzzword. These problems, of course, did not start with the Web. They have existed since designers began building user interface

Some Practical Measures of Usability

- Are people asking a lot of questions or often reaching for a manual?
- Are frequent exasperation responses heard?

- Are there many irrelevant actions being performed?
- Are there many things to ignore?
- Do a number of people want to use the product?

Some Objective Measures of Usability

Shackel (1991) presents the following more objective criteria for measuring usability.

- How effective is the interface? Can the required range of tasks be accomplished:
  - At better than some required level of performance (for example, in terms of speed and errors)?
  - By some required percentage of the specified target range of users?
  - Within some required proportion of the range of usage environments?
- How learnable is the interface? Can the interface be learned:
  - Within some specified time from commissioning and start of user training?
  - Based on some specified amount of training and user support?
  - Within some specified relearning time each time for intermittent users?
- How flexible is the interface? Is it flexible enough to:
  - Allow some specified percentage variation in tasks and/or environments beyond those first specified?
- What are the attitudes of the users? Are they:
  - Within acceptable levels of human cost in terms of tiredness, discomfort, frustration, and personal effort?
  - Such that satisfaction causes continued and enhanced usage of the system?

Human performance goals in system use, like any other design goal, should be stated in quantitative and measurable ways. Without performance goals you will never know if you have achieved them, or how successful the system really is. Clear and concrete goals also provide objectives for usability testing and ensure that a faulty or unsatisfactory product will not be released. The level of established goals will depend on the capabilities of the user, the capabilities of the system, and the objectives of the system

The Design Team

Provide a balanced design team, including specialists in:

- Development
- Human factors
- Visual design
- Usability assessment
- Documentation
- Training

Effective design and development requires the application of very diverse talents. A balanced design team with very different talents must be established. Needed are specialists in development to define requirements and write the software, human factors specialists to define behavioural requirements and apply behavioural considerations, and people with good visual design skills. Also needed are people skilled in testing and usability assessment, documentation specialists, and training specialists. Also, select team members who can effectively work and communicate with one another. To optimize communication, locate the team members in close proximity to one another.

CHAPTER 4.

KNOW YOUR USER OR CLIENT

Important Human Characteristics in Design

We are complex organisms with a variety of attributes that have an important influence on interface and screen design

Perception: Perception is our awareness and understanding of the elements and objects of our environment through the physical sensation of our various senses, including sight, sound, smell, and so forth. Perception is influenced, in part, by experience.

Other perceptual characteristics include the following:

- **Proximity:** Our eyes and mind see objects as belonging together if they are near each other in space.
- **Similarity:** Our eyes and mind see objects as belonging together if they share a common visual property, such as colour, size, shape, brightness, or orientation.
- **Matching patterns:** We respond similarly to the same shape in different sizes. The letters of the alphabet, for example, possess the same meaning, regardless of physical size.
- **Succinctness:** We see an object as having some perfect or simple shape because perfection or simplicity is easier to remember.
- **Closure:** Our perception is synthetic; it establishes meaningful wholes. If something does not quite close itself, such as a circle, square, triangle, or word, we see it as closed anyway.
- **Unity:** Objects that form closed shapes are perceived as a group.
- **Continuity:** Shortened lines may be automatically extended.
- **Balance:** We desire stabilization or equilibrium in our viewing environment. Vertical, horizontal, and right angles are the most visually satisfying and easiest to look at.
- **Expectancies:** Perception is also influenced by expectancies; sometimes we perceive not what is there but what we expect to be there. Missing a spelling mistake in

proofreading something we write is often an example of a perceptual expectancy error; we see not how a word is spelled, but how we expect to see it spelled.

- **Context:** Context, environment, and surroundings also influence individual perception. For example, two drawn lines of the same length may look the same length or different lengths, depending on the angle of adjacent lines or what other people have said about the size of the lines.
- **Signals versus noise:** Our sensing mechanisms are bombarded by many stimuli, some of which are important and some of which are not. Important stimuli are called signals; those that are not important or unwanted are called noise. The elements of a screen assume the quality of signal or noise, depending on the actions and thought processes of the user. Once a screen is first presented and has to be identified as being the correct one, the screen's title may be the signal, the other elements it contains simply being noise. When the screen is being used, the data it contains becomes the signal, and the title now reverts to noise. Other elements of the screen rise and fall in importance, assuming the roles of either signals or noise, depending on the user's needs of the moment. The goal in design is to allow screen elements to easily assume the quality of signal or noise, as the needs and tasks of the user change from moment to moment.

The goal in design, then, is to utilize our perceptual capabilities so a screen can be structured in the most meaningful and obvious way

Memory

Memory is viewed as consisting of two components, long-term and short-term (or working) memory.

- Short-term, or working, memory receives information from either the senses or long-term memory, but usually cannot receive both at once, the senses being processed separately. Within short-term memory a limited amount of information processing takes place. Information stored within it is variously thought to last from 10 to 30 seconds, with the lower number being the most reasonable speculation. Knowledge, experience, and familiarity govern the size and complexity of the information that can be remembered. Long-term memory contains the knowledge we possess. Information received in short-term memory is transferred to it and encoded within it, a process we call learning. It is a complex process requiring some effort on our part. The learning process is improved if the information being transferred from short-term memory has structure and is meaningful and familiar. Learning is also improved through repetition. Unlike short-term memory, with its distinct limitations, long-term memory capacity is thought to be unlimited. An important memory consideration, with significant implications for interface design, is the difference in ability to recognize or recall words.

Sensory Storage

Sensory storage is the buffer where the automatic processing of information collected from our senses takes place. It is an unconscious process, large, attentive to the environment, quick to detect changes, and constantly being replaced by newly gathered stimuli. In a sense, it acts like radar, constantly scanning the environment for things that are important to pass on to higher memory. Repeated and excessive stimulation can fatigue the sensory storage mechanism, making it less attentive and unable to distinguish what is important (called habituation). Avoid unnecessarily stressing it. Design the interface so that all aspects and elements serve a definite purpose. Eliminating interface noise will ensure that important things will be less likely to be missed.

Visual Acuity

The capacity of the eye to resolve details is called visual acuity. It is the phenomenon that results in an object becoming more distinct as we turn our eyes toward it and rapidly losing distinctness as we turn our eyes away—that is, as the visual angle from the point of fixation increases. It has been shown that relative visual acuity is approximately halved at a distance of 2.5 degrees from the point of eye fixation. The eye's sensitivity increases for those characters closest to the fixation point ("0") and decreases for those characters at the extreme edges of the circle (a 50/50 chance exists for getting these characters correctly identified). This may be presumed to be a visual ―chunk of a screen.

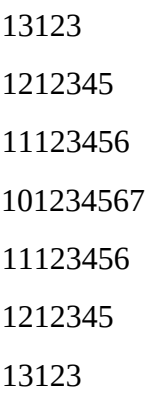


Figure 1.1 Size of area of optimum visual acuity on a scree

Foveal and Peripheral Vision

Foveal vision is used to focus directly on something; peripheral vision senses anything in the area surrounding the location we are looking at, but what is there cannot be clearly resolved because of the limitations in visual acuity just described. Foveal and peripheral vision maintain, at the same time, a cooperative and a competitive relationship. Peripheral vision can aid a visual search, but can also be distracting. In its cooperative nature, peripheral vision is thought to provide clues to where the eye should go next in the visual search of a screen. In its competitive nature, peripheral vision can compete with foveal vision for attention. What is sensed in the

periphery is passed on to our information processing system along with what is actively being viewed foveally.

Information Processing

The information that our senses collect that is deemed important enough to do something about then has to be processed in some meaningful way. There are two levels of information processing going on within us. One level, the highest level, is identified with consciousness and working memory. It is limited, slow, and sequential, and is used for reading and understanding. In addition to this higher level, there exists a lower level of information processing, and the limit of its capacity is unknown. This lower level processes familiar information rapidly, in parallel with the higher level, and without conscious effort. Both levels function simultaneously, the higher level performing reasoning and problem solving, the lower level perceiving the physical form of information sensed.

When a screen is displayed, you usually will want to verify that it is the one you want. If you're new to a system, or if a screen is new to you, you rely on its concrete elements to make that determination, its title, the controls and information it contains, and so forth. You consciously look at the screen and its components using this higher-level processing.

If a screen is jammed with information and cluttered, it loses its uniqueness and can only be identified through the more time-consuming, and thought-interrupting, reading process.

Mental Models

A mental model is simply an internal representation of a person's current understanding of something. Usually a person cannot describe this mental mode and most often is unaware it even exists. Mental models are gradually developed in order to understand something, explain things, make decisions, do something, or interact with another person. Mental models also enable a person to predict the actions necessary to do things if the action has been forgotten or has not yet been encountered. A person already familiar with one computer system will bring to another system a mental model containing specific visual and usage expectations. If the new system complies with already-established models, it will be much easier to learn and use. The key to forming a transferable mental model of a system is design consistency and design standards.

Movement Control

Once data has been perceived and an appropriate action decided upon, a response must be made; in many cases the response is a movement. In computer systems, movements include such activities as pressing keyboard keys, moving the screen pointer by pushing a mouse or rotating a trackball, or clicking a mouse button. Particularly important in screen design is Fitts' Law (1954). This law states that:

- The time to acquire a target is a function of the distance to and size of the target.

This simply means that the bigger the target is, or the closer the target is, the faster it will be reached. The implications in screen design are:

- Provide large objects for important functions.
- Take advantage of the ―pinning actions of the sides, top, bottom, and corners of the screen.

Learning

Learning, as has been said, is the process of encoding in long-term memory information contained in short term memory. A design developed to minimize human learning time can greatly accelerate human performance. People prefer to stick with what they know, and they prefer to jump in and get started that is contained in short-term memory.

Learning can be enhanced if it:

- Allows skills acquired in one situation to be used in another somewhat like it. Design consistency accomplishes this.
- Provides complete and prompt feedback.
- Is phased, that is, it requires a person to know only the information needed at that stage of the learning process.

Skill

The goal of human performance is to perform skill-fully. To do so requires linking inputs and responses into a sequence of action. The essence of skill is performance of actions or movements in the correct time sequence with adequate precision. Skills are hierarchical in nature, and many basic skills may be integrated to form increasingly complex ones. Lower-order skills tend to become routine and may drop out of consciousness.

Individual Differences

In reality, there is no average user. A complicating but very advantageous human characteristic is that we all differ—in looks, feelings, motor abilities, intellectual abilities, learning abilities and speed, and so on. Individual differences complicate design because the design must permit people with widely varying characteristics to satisfactorily and comfortably learn the task or job, or use the Web site. Multiple versions of a system can easily be created. Design must provide for the needs of all potential users.

Human Interaction Speeds

The speed at which people can perform using various communication methods has been studied by a number of researchers. The following, are summarized as table below

Reading

Prose text: 250–300 words per minute.

Proofreading text on paper: 200 words per minute.

Proofreading text on a monitor: 180 words per minute.

Listening: 150–160 words per minute.

Speaking to a computer: 105 words per minute.

After recognition corrections: 25 words per minute.

Keying

a. Typewriter
   Fast typist: 150 words per minute and higher.
   Average typist: 60–70 words per minute.
b. Computer
   Transcription: 33 words per minute.
   Composition: 19 words per minute.
c. Two finger typists
   Memorized text: 37 words per minute.
   Copying text: 27 words per minute.

Hand printing

Memorized text: 31 words per minute.

Copying text: 22 words per minute.

CHAPTER 5

UNDERSTAND THE BUSINESS FUNCTION

The general steps to be performed are:

- Perform a business definition and requirements analysis.
- Determine basic business functions.
- Describe current activities through task analysis.
- Develop a conceptual model of the system.
- Establish design standards or style guides.
- Establish system usability design goals.
- Define training and documentation needs.

Business Definition and Requirements Analysis

The objective of this phase is to establish the need for a system. A requirement is an objective that must be met. A product description is developed and refined, based on input from users or marketing. There are many techniques for capturing information for determining Business requirements. Keil and Carmel, Popowicz and Fuccella described methods as Direct and Indirect methods.

DIRECT METHODS

Advantages

- The significant advantage of the direct methods is the opportunity they provide to hear the user's comments in person and firsthand.
- Person-to-person encounters permit multiple channels of communication (body language, voice inflections, and so on) and provide the opportunity to immediately follow up on vague or incomplete data.

Here are some recommended direct methods for getting input from users.

Individual Face-to-Face Interview

A one-on-one visit with the user to obtain information. It may be structured or somewhat open-ended. A formal questionnaire should not be used, however. Useful topics to ask the user to describe in an interview include:

- The activities performed in completing a task or achieving a goal or objective.
- The methods used to perform an activity.
- What interactions exist with other people or systems?

It is also very useful to also uncover any:

- Potential measures of system usability.
- Unmentioned exceptions to standard policies or procedures.

- Relevant knowledge the user must possess to perform the activity.

If designing a Web site, the following kinds of interview questions are appropriate for asking potential users:

- Present a site outline or proposal and then solicit comments on the thoroughness of content coverage, and suggestions for additional content.
- Ask users to describe situations in which the proposed Web site might be useful.
- Ask users to describe what is liked and disliked about the Web sites of potential competitors.
- Ask users to describe how particular Web site tasks should be accomplished.

Advantages

Advantages of a personal interview are that you can give the user your full attention, can easily include follow-up questions to gain additional information, will have more time to discuss topics in detail, and will derive a deeper understanding of your users, their experiences, attitudes, beliefs, and desires.

Disadvantages

Disadvantages of interviews are that they can be costly and time consuming to conduct, and someone skilled in interviewing techniques should perform them.

Telephone Interview or Survey

A structured interview conducted via telephone.

Advantages

- Arranging the interview in advance allows the user to prepare for it.
- Telephone interviews are less expensive and less invasive than personal interviews.
- They can be used much more frequently and are extremely effective for very specific information.

Disadvantage

- It is impossible to gather contextual information, such as a description of the working environment, replies may be easily influenced by the interviewer's comments, and body language cues are missing.
- Also, it may be difficult to contact the right person for the telephone interview.

Traditional Focus Group

A small group of users and a moderator brought together to verbally discuss the requirements. The purpose of a focus group is to probe user's experiences, attitudes, beliefs, and desires, and to obtain their reactions to ideas or prototypes.

Setting up focus group involves the following:

- Establish the objectives of the session.

- Select participants representing typical users, or potential users.
- Write a script for the moderator to follow.
- Find a skilled moderator to facilitate discussion, to ensure that the discussion remains focused on relevant topics, and to ensure that everyone participates.
- Allow the moderator flexibility in using the script.
- Take good notes, using the session recording for backup and clarification.

Facilitated Team Workshop

A facilitated, structured workshop held with users to obtain requirements information. Similar to the traditional Focus Group. Like focus groups, they do require a great deal of time to organize and run.

Observational Field Study

Users are observed and monitored for an extended time to learn what they do. Observation provides good insight into tasks being performed, the working environment and conditions, the social environment, and working practices. Observation, however, can be time-consuming and expensive. Video recording of the observation sessions will permit detailed task analysis.

Requirements Prototyping

A demo, or very early prototype, is presented to users for comments concerning functionality.

User-Interface Prototyping

A demo, or early prototype, is presented to users to uncover user-interface issues and problems

Usability Laboratory Testing

Users at work are observed, evaluated, and measured in a specially constructed laboratory to establish the usability of the product at that point in time. Usability tests uncover what people actually do, not what they think they do a common problem with verbal descriptions. The same scenarios can be presented to multiple users, providing comparative data from several users. Problems uncovered may result in modification of the requirements. Usability labs can generate much useful information but are expensive to create and operate.

Card Sorting for Web Sites

A technique to establish groupings of information for Web sites. It is normally used only after gathering substantial site content information. Potential content topics are placed on individual index cards and users are asked to sort the cards into groupings that are meaningful to them. Card sorting assists in building the site's structure, map, and page content.

Briefly, the process is as follows:

- From previous analyses, identify about 50 content topics and inscribe them on index cards. Limit topics to no more than 100.

- Provide blank index cards for names of additional topics the participant may want to add, and coloured blank cards for groupings that the participant will be asked to create.
- Number the cards on the back.
- Arrange for a facility with large enough table for spreading out cards.
- Select participants representing a range of users. Use one or two people at a time and 5 to 12 in total.
- Explain the process to the participants, saying that you are trying to determine what categories of information will be useful, what groupings make sense, and what the groupings should be called.
- Ask the participants to sort the cards and talk out loud while doing so. Advise the participants that additional content cards may be named and added as they think necessary during the sorting process.
- Observe and take notes as the participants talk about what they are doing. Pay particular attention to the sorting rationale.
- Upon finishing the sorting, if a participant has too many groupings ask that they be arranged hierarchically.
- Ask participants to provide a name for each grouping on the coloured blank cards, using words that the user would expect to see that would lead them to that particular grouping.
- Make a record of the groupings using the numbers on the back of each card.
- Reshuffle the cards for the next session.
- When finished, analyze the results looking for commonalities among the different sorting sessions.

INDIRECT METHODS

An indirect method of requirements determination is one that places an intermediary between the developer and the user. This intermediary may be electronic or another person.

Problems of Indirect Method

- First, there may be a filtering or distortion of the message, either intentional or unintentional.
- Next, the intermediary may not possess a complete, or current, understanding of user's needs, passing on an incomplete or incorrect message.
- Finally, the intermediary may be a mechanism that discourages direct user-developer contact for political reasons.

MIS Intermediary

A company representative defines the user's goals and needs to designers and developers. This representative may come from the Information Services department itself, or he or she may be from the using department.

Paper Survey or Questionnaire

A survey or questionnaire is administered to a sample of users using traditional mail methods to obtain their needs.

Advantage

- Questionnaires have the potential to be used for a large target audience located most anywhere, and are much cheaper than customer visits.
- They generally, however, have a low return rate.

Disadvantage

- They may take a long time to collect and may be difficult to analyse.
- Questionnaires should be composed mostly of closed questions (yes/no, multiple choice, short answer, and so on).
- Questionnaires should be relatively short and created by someone experienced in their design.

Electronic Survey or Questionnaire

A survey or questionnaire is administered to a sample of users using e-mail or the Web to obtain their needs. In creating an electronic survey:

- Determine the survey objectives.
- Determine where you will find the people to complete the survey.
- Create a mix of multiple choice and open-ended questions requiring short answers addressing the survey objectives.
- Keep it short, about 10 items or less is preferable.
- Keep it simple, requiring no more than 5–10 minutes to complete

Iterative survey

- Consider a follow-up more detailed survey, or surveys, called iterative surveys. Ask people who complete and return the initial survey if they are willing to answer more detailed questions. If so, create and send the more detailed survey.
- A third follow-up survey can also be designed to gather additional information about the most important requirements and tasks.
- Iterative surveys, of course, take a longer time to complete.

Electronic Focus Group

A small group of users and a moderator discuss the requirements online using workstations.

Advantages.

- Advantages of electronic focus groups over traditional focus groups are that the discussion is less influenced by group dynamics; has a smaller chance of being dominated by one or a few participants; can be anonymous, leading to more honest comments and less caution in proposing new ideas; can generate more ideas in a shorter time since all participants can communicate at once; and can lead to longer sessions since the participant is in a more comfortable "home environment" and not confined to a conference room.

Disadvantages

- The depth and richness of verbal discussions does not exist and the communication enhancement aspects of seeing participant's body language are missing.

Marketing and Sales

Company representatives who regularly meet customers obtain suggestions or needs, current and potential.

Support Line

Information collected by the unit that helps customers with day-to-day problems is analysed (Customer Support, Technical Support, Help Desk, etc.). This is fairly inexpensive and the target user audience is correct. The focus of this method is usually only on problems, however.

E-Mail or Bulletin Board

Problems, questions, and suggestions from users posted to a bulletin board or through e-mail are analysed. Fairly inexpensive method and focuses only on problems

User Group

Improvements are suggested by customer groups who convene periodically to discuss software usage. They require careful planning. User groups have the potential to provide a lot of good information, if organized properly.

Competitor Analyses

A review of competitor's products or Web sites is used to gather ideas, uncover design requirements and identify tasks.

Trade Show

Customers at a trade show are presented a mock-up or prototype and asked for comments.

Other Media Analysis

An analysis of how other media, print or broadcast, present the process, information, or subject matter of interest. Findings can be used to gather ideas, uncover design requirements, and identify better ways to accomplish or show something.

System Testing

New requirements and feedback are obtained from ongoing product testing g can be accumulated, evaluated, and implemented as necessary.

Requirements Collection Guidelines

Keil and Carmel (1995) evaluated the suitability and effectiveness of various requirements-gathering method. Each requirements collection method was defined as a developer-user link. Their findings and conclusions

- Establish 4 to 6 different developer-user links:
  The more successful projects had utilized a greater number of developer-user links than the less successful projects. The mean number of links for the successful projects:5.4 the less successful: 3.2. Keil and Carmel recommend, based upon their data, that, at minimum, four different developer-user links must be established in the requirements-gathering process. They also concluded that the law of diminishing returns begins to set in after six links.
  Effectiveness ratings of the most commonly used links in their study were also obtained. (Not all the above-described techniques were considered in their study.) On a 1 to 5 scale (1 = ineffective, 5 = very effective) the following methods had the highest ratings:
  Custom projects (software developed for internal use and usually not for sale):

Facilitated Teams	5.0
User-Interface Prototype	4.0
Requirements Prototype	3.6
Interviews	3.5

  Package projects (software developed for external use and usually for sale):

Support Line	4.3
Interviews	3.8
User-Interface Prototype	3.3
User Group	3.3

- Provide the Most Reliance on Direct Links:
  The problems associated with the less successful projects resulted, at least in part, from too much reliance on indirect links, or using intermediaries. Ten of the 14 less successful projects had used none, or only one, direct link. The methods with the highest effectiveness ratings listed above were mostly direct links. Keil and Carmel caution that number of links is only a partial measure of user participation. How well the link is employed in practice is also very important.

Determining Basic Business Functions

A detailed description of what the product will do is prepared. Major system functions are listed and described, including critical system inputs and outputs. A flowchart of major functions is developed. The process the developer will use is summarized as follows:

- Gain a complete understanding of the user's mental model based upon:
  - The user's needs and the user's profile.
  - A user task analysis.
- Develop a conceptual model of the system based upon the user's mental model. This includes:
  - Defining objects.
  - Developing metaphors.

Understanding the User's Mental Model

A goal of task analysis, and a goal of understanding the user, is to gain a picture of the user's mental model. A mental model is an internal representation of a person's current conceptualization and understanding of something. Mental models are gradually developed in order to understand, explain, and do something. Mental models enable a person to predict the actions necessary to do things if the actions have been forgotten or have not yet been encountered.

Performing a Task Analysis

User activities are precisely described in a task analysis. Task analysis involves breaking down the user's activities to the individual task level. The goal is to obtain an understanding of why and how people currently do the things that will be automated.

Knowing why establishes the major work goals; knowing how provides details of actions performed to accomplish these goals. Task analysis also provides information concerning workflows, the interrelationships between people, objects, and actions, and the user's conceptual frameworks. The output of a task analysis is a complete description of all user tasks and interactions.

One result of a task analysis is a listing of the user's current tasks. This list should be well documented and maintained. Changes in task requirements can then be easily incorporated as design iteration occurs. Another result is a list of objects the users see as important to what they do. The objects can be sorted into the following categories:

- Concrete objects—things that can be touched.
- People who are the object of sentences—normally organization employees, customers, for example.
- Forms or journals—things that keep track of information.
- People who are the subject of sentences—normally the users of a system.
- Abstract objects—anything not included above.

Developing Conceptual Models

The output of the task analysis is the creation, by the designer, of a conceptual model for the user interface. A conceptual model is the general conceptual framework through which the system's functions are presented. Such a model describes how the interface will present objects, the relationships between objects, the properties of objects, and the actions that will be performed. The goal of the designer is to facilitate for the user the development of useful mental model of the system. This is accomplished by presenting to the user a meaningful conceptual model of the system. When the user then encounters the system, his or her existing mental model will, hopefully, mesh well with the system's conceptual model.

Guidelines for Designing Conceptual Models

- **Reflect the user's mental model not the designer's:** A user will have different expectations and levels of knowledge than the designer. So, the mental models of the user and designer will be different. The user is concerned with the task to be performed, the business objectives that must be fulfilled.
- **Draw physical analogies or present metaphors:** Replicate what is familiar and well known. Duplicate actions that are already well learned. A metaphor, to be effective, must be widely applicable within an interface.
- **Comply with expectancies, habits, routines, and stereotypes:** Use familiar associations, avoiding the new and unfamiliar. With colour, for example, accepted meanings for red, yellow, and green are already well established. Use words and symbols in their customary ways.
- **Provide action-response compatibility:** All system responses should be compatible with the actions that elicit them. Names of commands, for example, should reflect the actions that will occur.
- **Make invisible parts and process of a system visible:** New users of a system often make erroneous or incomplete assumptions about what is invisible and develop a faulty mental model. As more experience is gained, their mental models evolve to become more accurate and complete. Making invisible parts of a system visible will speed up the process of developing correct mental models. In a graphical direct manipulation system, the entire process is visible, with the user literally picking up the file in one folder by clicking on it, and dragging it to another folder.
- **Provide proper and correct feedback:** Be generous in providing feedback. Keep a person informed of what is happening, and what has happened, at all times, including:

    - Provide a continuous indication of status: During long processing sequences, for example, interim status messages such as loading, "opening . . ." or "searching . . ." reassure the user and enable him or her to understand internal processes and more accurately predict how long something will take. Such messages also permit the pinpointing of problems if they occur

  - Provide visible results of actions: For example, highlight selected objects, display new locations of moved objects, and show files that are closed.
  - Display actions in progress: For example, show a window that is being changed in size actually changing, not simply the window in its changed form.
  - Present as much context information as possible: For example, on a menu screen or in navigation, maintain a listing of the choices selected to get to the current point. On a query or search screen, show the query or search criteria when displaying the results
  - Provide clear, constructive, and correct error messages: For example, error messages addressing an incomplete action should specify exactly what is missing, not simply advise a person that something is incomplete
- **Avoid anything unnecessary or irrelevant:** Never display irrelevant information on the screen. People may try to interpret it and integrate it into their mental models, thereby creating a false one.
- **Provide design consistency:** Design consistency reduces the number of concepts to be learned. Inconsistency requires the mastery of multiple models. If an occasional inconsistency cannot be avoided, explain it to the user.
- **Provide documentation and a help system that will reinforce the conceptual model:** Do not rely on the people to uncover consistencies and metaphors themselves. The help system should offer advice aimed at improving mental models.
- **Promote the development of both novice and expert mental models:** Novices and experts are likely to bring to bear different mental models when using a system.

Defining Objects

- Determine all objects that have to be manipulated to get work done. Describe:
  - The objects used in tasks.
  - Object behaviour and characteristics that differentiate each kind of object.
  - The relationship of objects to each other and the people using them.
  - The actions performed.
  - The objects to which actions apply.
  - State information or attributes that each object in the task must preserve, display, or allow to be edited.
- Identify the objects and actions that appear most often in the workflow.
- Make the several most important objects very obvious and easy to manipulate.

Developing Metaphors

A metaphor is a concept where one's body of knowledge about one thing is used to understand something else. Metaphors act as building blocks of a system, aiding understanding of how a system works and is organized. Real-world metaphors are most often the best choice.

Replicate what is familiar and well known. A common metaphor in a graphical system is the desktop and its components,

- Choose the analogy that works best for each object and its actions.
- Use real-world metaphors.
- Use simple metaphors.
- Use common metaphors.
- Multiple metaphors may coexist.
- Use major metaphors, even if you can't exactly replicate them visually.
- Test the selected metaphors.

The User's New Mental Model

When the system is implemented, and a person interacts with the new system and its interface, an attempt will be made by the person to understand the system based upon the existing mental model brought to the interaction. If the designer has correctly reflected the user's mental model in design, the user's mental model is reinforced and a feeling that the interface is intuitive will likely develop. Continued interaction with the system may influence and modify the user's concept of the system, and his or her mental model may be modified as well. Refinement of this mental model, a normal process, is aided by well-defined distinctions between objects and being consistent across all aspects of the interface.

If the new system does not accurately reflect the user's existing mental model, then the results include breakdowns in understanding, confusion, errors, loss of trust, and frustration. Another result is an inability to perform the task or job.

Historically, when system designers have known in advance there was a gap between their conceptual model and the mental model the user would bring to the new system, designers have tried to bridge this gap through extensive documentation and training. The problems with this approach are: people are unproductive while being trained, people rarely read the documentation and training materials, and, even if the training material is read, the material is presented out of context. This creates difficulties for the users in understanding the material's relevance to their needs and goals

Design Standards or Style Guides

A design standard or style guide documents an agreed-upon way of doing something. It also defines the interface standards, rules, guidelines, and conventions that must be followed in detailed design.

Value of Standards and Guidelines

- Developing and applying design standards or guidelines achieve design consistency.
- This is valuable to users because the standards and guidelines:
    - Allow faster performance.
    - Reduce errors.

  - Reduce training time.
  - Foster better system utilization.
  - Improve satisfaction.
  - Improve system acceptance.
- They are valuable to system developers because they:
  - Increase visibility of the human-computer interface.
  - Simplify design.
  - Provide more programming and design aids, reducing programming time.
  - Reduce redundant effort.
  - Reduce training time.
  - Provide a benchmark for quality control testing.

Document Design

- Include checklists to present principles and guidelines.
- Provide a rationale for why the particular guidelines should be used.
- Provide a rationale describing the conditions under which various design alternatives are appropriate.
- Include concrete examples of correct design.
- Design the guideline document following recognized principles for good document design.
- Provide good access mechanisms such as a thorough index, a table of contents, glossaries, and checklists.

Checklists and rationale: Provide checklists for presenting key principles and guidelines. Checklists permit ease in scanning, ease in referring to key points, and make a document more readable by breaking up long sequences of text. Also provide a rationale for why the particular guidelines should be used. Understanding the reasoning will increase guideline acceptance. This is especially important if the guideline is a deviation from a previous design practice. Also, when two or more design alternatives exist, provide a rationale describing the conditions under which the alternatives are appropriate. It may not be easy for designers to infer when various alternatives are appropriate. You have probably noticed that this book uses a checklist format to present key guidelines and thoughts, and guideline rationale is described in the text.

Concrete examples: To be effective, a guideline must include many concrete examples of correct design. Learning by imitation is often a way we learn.

Document design and access: Always design the document, be it paper or electronic, by following recognized principles for good document design. This greatly enhances readability. Provide good access mechanisms such as a thorough index, a table of contents, glossaries, and checklists. An unattractive or hard to use document will not be inviting and consequently will not be used

Design Support and Implementation

- Use all available reference sources in creating the guidelines.
- Use development and implementation tools that support the guidelines.
- Begin applying the guidelines immediately.

Available Reference Sources. Use all the available reference design sources in creating your guidelines. References include this text, other books on user interface design, project-specific guidelines, and the style guides for interface design and Web design created by companies such as Apple, IBM, Microsoft, and Sun. Other reference sources that meet your needs should also be utilized.

Tools: Use tools that support implementation of the guidelines you have established. Development tools make the design process much easier. If the design tools cannot support the guideline, it cannot be adhered to.

Applying the Guidelines: researchers reformatted several alphanumeric inquiry screens to improve their comprehensibility and readability. When these reformatted screens were presented to expert system users, decisionmaking time remained the same but errors were reduced. For novice system users, the reformatted screens brought large improvements in learning speed and accuracy. Therefore, it appears, that changes that enhance screens will benefit both novice and expert users already familiar with the current screens. It is never too late to begin to change

MODULE 3

CHAPTER 6. DEVELOP SYSTEM MENUS AND NAVIGATION SCHEMES

Syllabus: Structures of Menus, Functions of Menus, Content of Menus, Formatting of Menus, Phrasing the Menu, Selecting Menu Choices, Navigating Menus, Kinds of Graphical Menus.

A system contains large amounts of information and performs a variety of functions. 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.

Menus are effective because they utilize the more powerful human capability of recognition rather than the weaker capability of recall. 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.

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 level 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:

1. Single Menus
2. Sequential Linear Menus
3. Simultaneous Menus
4. Hierarchical Menus
5. Connected Menus
6. Event Trapping Menus

Single Menus

- 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 3.1. 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.

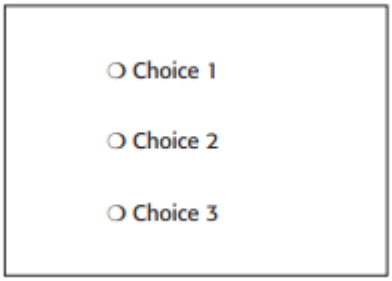


Figure 3.1: Single Menu

- 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 re-entry 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 pre-set 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.

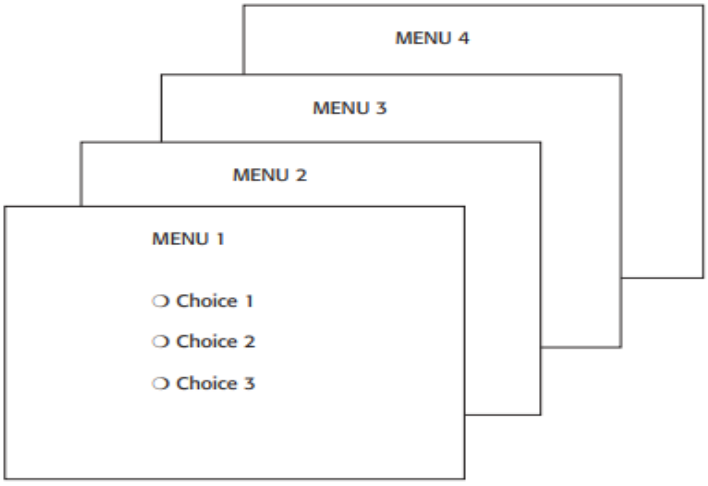


Figure 3.2: Sequential linear menus

- Sequential path menus have several shortcomings. A long sequence may become tedious as menu after menu is presented. Example: 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 which they are being presented.

Simultaneous Menus

- Instead of being presented on separate screens, all menu options are available simultaneously.
- 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.

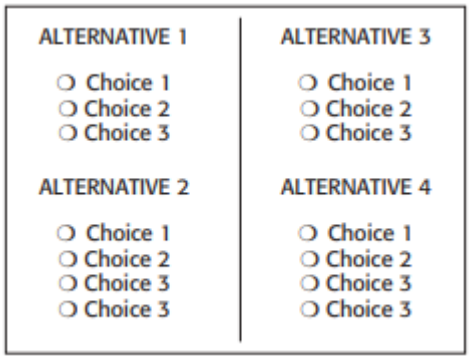


Figure 3.3: Simultaneous Menus

- 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.
- Presenting many menu dependencies and relationships on a screen, especially if poorly indicated, can also be very confusing

Hierarchical Menus

- A hierarchical structure results in an increasing refinement of choice as menus are stepped through, for example, from options, to sub-options, 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.
- Common examples of hierarchical design today are found in menu bars with their associated pull-downs.
- 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.

- A disadvantage of a hierarchical scheme is that the defined branching order may not fit the users conception of the task flow
- If users are not familiar with the hierarchical menu, or are unable to predict what sub-options 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 must also be easy to back upward through the tree to facilitate exploration of the tree. 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.

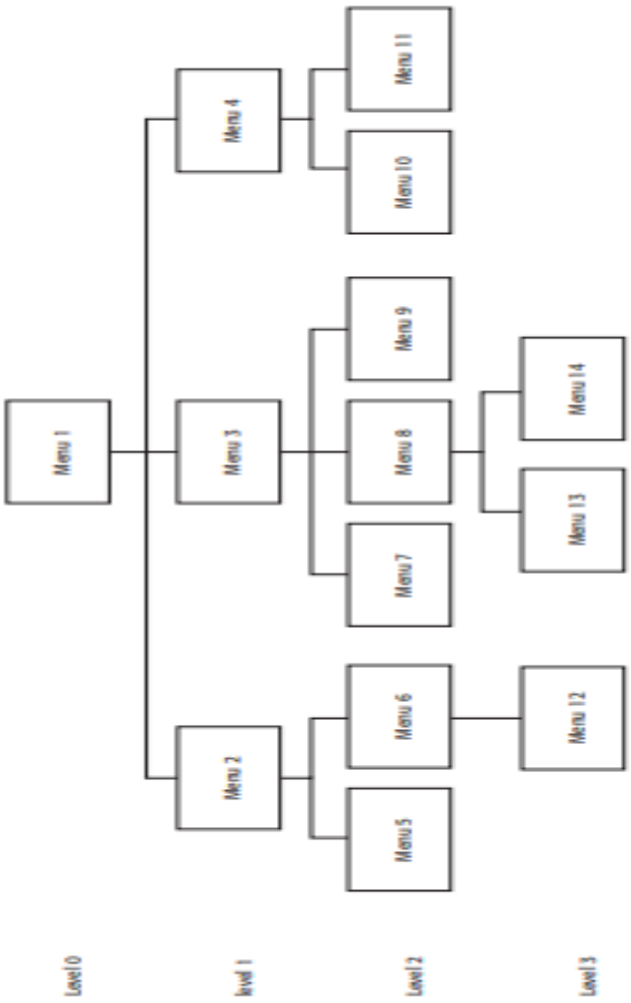


Figure 3.4: Hierarchical Menus

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.
- 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.
- 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.

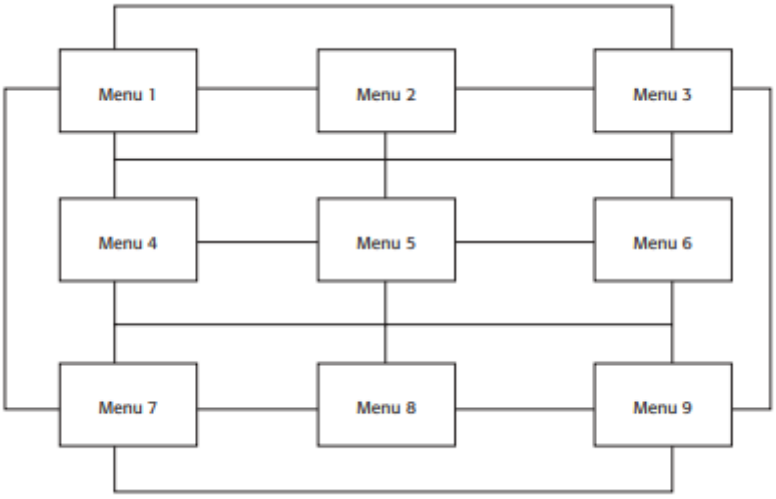


Figure 3.5: Connected Menus

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.
- 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

A menu can be used to perform several functions:

1. Navigate to a new menu
2. Execute an action or procedure
3. Displaying information
4. 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 non-destructive 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.
- 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 content material and the user's interests will determine the paths followed.
- The user's focus is primarily on the information desired and less on the selection function. Wrong turns in the process will again cost time and perhaps aggravation, but these errors are non-destructive 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.

CONTENT OF MENUS

A menu consists of four elements:

1. Context
2. Title
3. Choice descriptions
4. Completion instructions.

Menu Context

- A menu's context provides information to keep the user oriented.
- 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.
- 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
- 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.

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.

Completion Instructions

- Completion instructions tell users how to indicate their choices
- 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

A series of guidelines for formatting menus are as follows

1. 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.

2. 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. [pop-ups or pull-downs]
    - Critical options should be continuously displayed, however.

3. Presentation

- Ensure that a menu and its choices are obvious [immediately recognizable] to the user by presenting them with a unique and consistent structure, location, and/or display technique. Example: 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. Web page navigation links, which may be scattered throughout a page, are displayed underlined and in a unique to differentiate and identify them
- Ensure that other system components do not possess the same visual qualities as menu choices.

4. Organization

- Provide a general or main menu. [Consisting of basic options. A starting point or "Home Base" to which user may always return]
- Display:
    - All relevant alternatives.
    - Only relevant alternatives - Delete or grey-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.
- Provide users with an easy way to restructure a menu according to how work is accomplished.

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 since 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 since 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.

5. Complexity

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

6. 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.

7. Ordering

- Order lists of choices by their natural order, [Months of a year, Days in a week]
- 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 [such as Delete or Clear] from frequently chosen items. [to minimize accidental selection]
- If option usage changes, do not reorder menus. [Dynamic menus were slower to use and less preferred than static menus. The continual reordering interfered with menu order learning]
- Maintain a consistent ordering of options on all related menus.
  - For variable-length menus, maintain consistent relative positions. [Place Exit at the bottom or end of the list]
  - For fixed-length menus, maintain consistent absolute positions.

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.

8. 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.

9. 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.

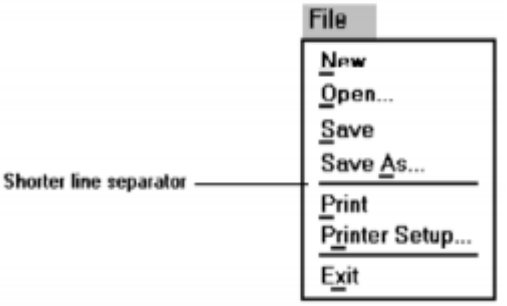


Figure 3.6: Partial line separators

- For independent groupings:
    - Extend the line to the left and right menu borders.

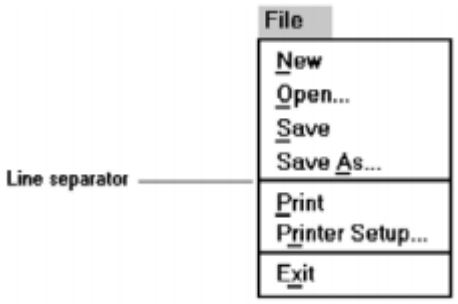


Figure 3.7: Extended line separators

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 called thoroughness. 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. 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.

1. 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 headline style.
  - Superfluous titles may be omitted. [Titles that add nothing to the understanding of menu content or context]

2. Menu Choice Descriptions

- Create meaningful choice descriptions that are familiar, fully spelled out, concise, and distinctive. [meaningful - “print” is more descriptive than “list”]
- 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 headline style, capitalizing the first letter of each significant word in the choice description.
- Use task-oriented not data-oriented wording. [Example-“Manage Customer Information” rather than “Customers”]
- Use parallel construction. [Example- Print a File, Execute a Program, Disk Eject]
- 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.

3. **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.
       • Left-justify the instruction and indent the related menu choice descriptions a minimum of three spaces to the right.
       • Leave a space line, if possible, between the instructions and the related menu choice descriptions.
     - 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 colour.

4. **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. [Enhances predictability and exploration of a graphical system]

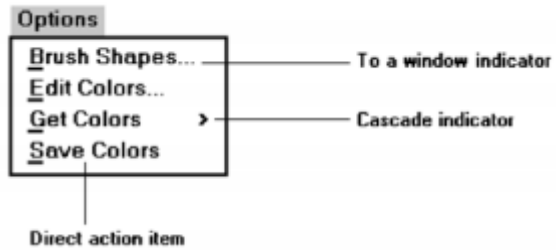


Figure 3.8: Intent Indicators

   - 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.

5. **Keyboard Equivalents**
   - To facilitate keyboard selection of a menu choice, each menu item should be assigned a keyboard equivalent mnemonic.

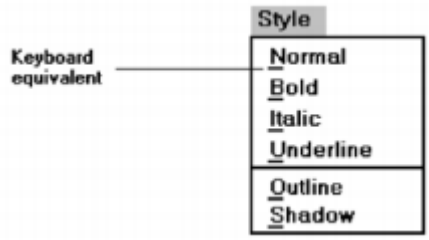


Figure 3.9: Keyboard Equivalents

- 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 [Access keys]

Table 4.1 Standard Keyboard Equivalents

About	Help	Print	Send To
Apply	Help Topics	Print Preview	Show
Back	Insert	Properties	Size
Browse	Maximize	Redo	Split
Close	Minimize	Repeat	Stop
Copy	Move	Restore	Undo
Cut	New	Resume	View
Delete	Next	Retry	Yes
Edit	No	Rerun	
Exit	Open	Save	
File	Paste	Save As	
Find	Page Setup	Select All	

A great deal of commonality exists among these equivalents since 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.

6. 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. [Shortcut keys also called Hot 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.

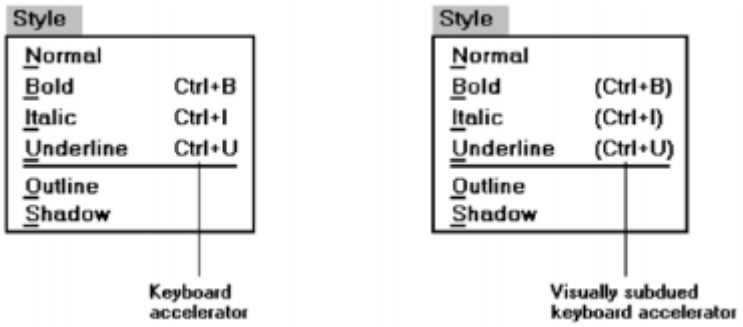


Figure 3.10: Keyboard Accelerators

- 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

Table 4.2 Standard Keyboard Accelerators

THIS ACCELERATOR:	DESIGNATES THIS ACTION:
Ctrl+C	Copy
Ctrl+N	New
Ctrl+0	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

1. **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.
2. **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 milli-meters 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 milli-meters 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. [Example: Require typing the mnemonic to select and then a press of the Enter or Return key to execute].
     - 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.

3. Defaults

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

4. 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.

5. 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.

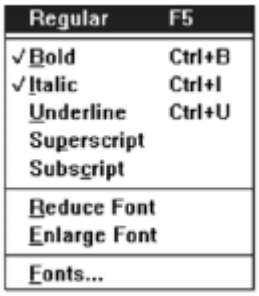


Figure 3.11: Mark Toggles

- 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.

6. 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.

Figure 3.12: Toggled menu item

- 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.

NAVIGATING MENUS

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. In Web site design, the most successful sites have been found to be those with easy to use and understand navigational systems. A system's organizational structure and its navigational tools, including elements such as menus, links, toolbars, and command buttons influence the system's navigational ease of use, including site maps and indexes. In Web site navigation design, the unique, often incompatible, relationship between the browser and the Web site application being presented can also strain navigation ease.

Web Site Navigation Problems

The Web and its navigation are undoubtedly the most complex interface facing people today.

1. Technical issues:
   - In a Graphical system application, whose screens tend to flow in an orderly and predictable manner, but a Web application is composed of pages, each of which can, theoretically, be linked to any other page in the application.
   - 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 can perform tasks or satisfy needs at will, easily moving between most screens in the application "spider web" in any order desired, and even jumping to other spider webs when the urge arises. [Example: In an analogy to driving a car, the graphical system user is essentially following a freeway in Nevada. The Web 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 redisplays the page that was previously displayed on the screen. This page may have been in another Web site, and the user is transported there.
   - 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, 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.

2. 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. The cognitive or mental overhead the user must expend in making decisions concerning which links to follow, or to abandon, can be overwhelming, if there are too many links presented on a page, many of whose meanings are not clear. Another problem is that not all links on a page are always obvious. This often leads to much trial-and-error behaviour; the user aimlessly clicking to see what happens. Feelings of disorientation are easily experienced when one becomes "lost in Web space."

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.

Navigation Goals

A well-designed navigation system facilitates quick and easy navigation between components whose structure and relationship are easily comprehendible. 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?

General system navigation guidelines include the following:

1. **Control**
   - For multilevel menus, provide one simple action to:
     - Return to the next higher-level menu.
     - Return to the main menu.
   - Provide multiple pathways through a menu hierarchy whenever possible.
2. **Menu Navigation Aids**
   - To aid menu navigation and learning, provide an easily accessible:
     - Menu map [Graphic representation of menu map] or overview of the menu hierarchy. [System documentation with Help function]
     - A "look ahead" at the next level of choices, alternatives that will be presented when a currently viewed choice is selected. [pull-down menus]
     - Navigation history. [Summarizes menu choices made]

Web Site Navigation

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 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. The potential for getting lost is extremely high, unless numerous, obvious, and understandable landmarks are available as a guide.

Web site navigation design involves

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 comprehendible 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: 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.

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, since 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.

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

Hierarchical tree: 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. 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.

Components of a Web Navigation System

To move between Web site information fragments necessitates the creation of navigation links. They are contained within a framework of tools or controls called components such as,

1. Browser's command buttons
2. Textual phrases
3. Web site navigation bars and Web site command buttons.

Collectively, these are all referred to as links.

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. links consisted of textual or binary files. Utilization of hypertext on the Web allowed links to by created using images as well as text, so the term hypermedia was coined to reflect this expanded nature.

General link guidelines are:

All navigation controls must:

- Make sense in the absence of site context.
- Be continually available.
- Be obvious and distinctive. [The obviousness of a link is called its affordance. A control with high affordance will be quickly identified as a control] [Distinctive- Tools appearance suggests user it is selected or not]
- Be consistent in appearance, function, and ordering.
- Possess a textual label or description.

- Offer multiple navigation paths. [Command Buttons such as *Next* and *Previous*]

1. **Browser Command Buttons**
   - Hide the split between the browser and the Web site application by including navigational controls within the application.

The browser being used in interacting with the Web provides its own command buttons. Pressing the browser Back and Forward buttons can create confusion because they can transport a user in and out of a Web site. Rather then relying on the browser's buttons, provide navigation controls within the application for movement within the application. They can take the form of links or command buttons such as Next and Previous.

2. **Web Site Navigation Bars**
   - Provide a global navigation bar at the top of each page. [site's total scope or categories of available information]
   - Provide a local category or topical links navigation bar on the left side of a page.
   - Place minor illustrative, parenthetical, or footnote links at the end of the page.
   - For long pages, provide a navigation bar repeating important global or local links at the page bottom.

A Web site navigation bar is a menu, an array of textual phrases, graphical images or icons, or command buttons, as illustrated in Figures 3.13, 3.14, and 3.15. A Web site contains at least three levels of navigation links, global, or site-wide. Display of a page and its content, a top-to-bottom results in sequential eye flow, but is quite wasteful of screen space, consuming about 20 percent of a screen's pixels. 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. Figure 3.16 separates navigation from content, making it easy for users to find each

Figure 3.13: Textual explicit listing navigation bars

Figure 3.14: Graphical or iconic navigational bars

Figure 3.15: Command button navigation bar

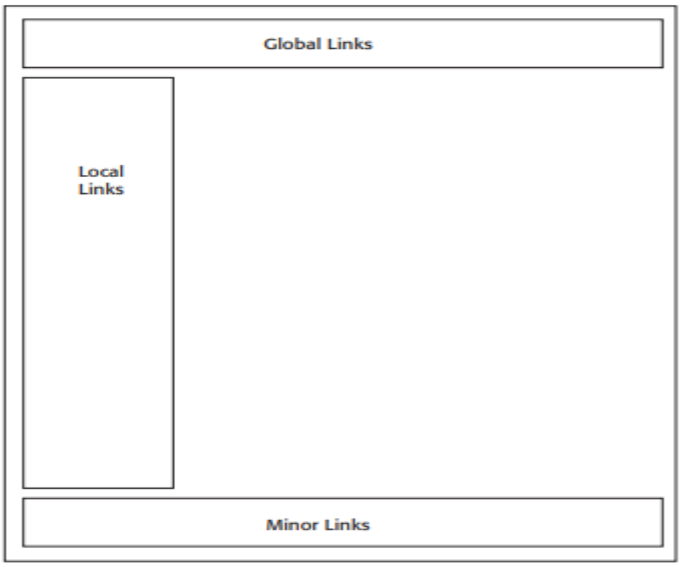


Figure 3.16: Web navigation component locations

3. Textual Phrases

- 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. 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 a toolbar. 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 4.13. 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 4.17. Web sites usually contain both explicit link listings and embedded links in various mixes.

Graphical Images or Icons

Graphical images or icons may appear in an array in the form of a navigation bar, or be individually located at relevant points in a page.

Command Buttons

Command buttons may appear in an array in the form of a navigation bar, or be individually located at relevant points in a page.

Other Web Site Navigation Elements

In addition to Navigation bars, a number of other Web site elements are also important components of the Web navigation system. Among these are

1. Overviews-including executive summaries, site maps, indexes, and tables of contents.
2. Historical trails
3. Search engines.

1. 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. [Through keywords]
     - A table of contents. [major topics and subtopics within]
   - 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.

2. Historical Trails
   - Provide:
     - Breadcrumb Trails.
       - Locate at the top of the page below the navigation links.
     - History Lists.

- History Trees.
- Footprints.
- Bookmarks.

Figure 3.17: Breadcrumb Trails

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 enables 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.

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 3.17, is also a series of links that permit the user to go back to any page in the sequence with one click. Symbols used include an arrow (->), a colon (:), a greater than sign (>), and a slash (/).

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 signalling 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

3. Search Facility
   - Provide a search facility.

Web Site Navigation Guidelines

1. Scrolling
   - Do not require scrolling of navigation-only pages.
   - Minimize the need for scrolling to view all links on pages containing content.
   - Never require horizontal scrolling. [makes text reading difficult]
2. Number of Links
   - Every page should contain at least one link.
   - Be conservative in the total number of links presented on a screen:
     - Without logical groupings of elements, limit links to 4 to 8.
     - With logical groupings of elements, limit links to 18 to 24.

- Restrict embedded links to those most important, pertinent, and interesting.
    - Place less relevant links in a listing.

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.

3. Presenting Links
    - Link text:
        - Underline all link text, including that:
            - Embedded in page content.
            - Contained in explicit menu listings.
            - Contained in headings.
            - Used as graphical labels.
        - Distinguish between unselected/unvisited links and selected/visited links.
            - Make unselected/unvisited links blue.
            - Make selected/visited links purple.
    - Kinds of links:
        - Distinguish links leading to different Web destinations through a differentiating symbol:
            - Precede links to content within the same page with a pound sign (#).
                - For links moving downward in the page, use: #The principles of design.
                - For links moving upward in the page use: #^Principles introduction.
            - Precede links to external or foreign sites with another unique symbol such as an asterisk (*): *Additional information.
            - Do not precede links to other site pages with any symbol:
                - More principles of design.
        - Also distinguish links leading to different Web destinations by presenting them in consistent locations.
    - Graphical links:
        - Distinguish graphical links from decorative graphics through:
            - Underlining graphical text labels.
    - Links in toolbars:
        - Distinguish links contained in toolbars through:
            - Presenting in consistent locations. [top of a page]
            - Using different coloured backgrounds.

Other Link Guidelines

1. Writing:

- Provide links to satisfy a range of user needs.
- Create descriptive links clearly indicating their destination or resulting action.

2. 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.
3. Ordering:
   - Order links by the most relevant menu choice-ordering scheme.
4. Heading:
   - Where appropriate, provide a listing heading describing the organizing category, principle, or theme.
5. Size:
   - Provide graphical images and command buttons of sufficient and equal size.
6. Spacing:
   - Create equal spacing between choices graphical image and textual listing toolbars.
7. Inapplicability:
   - Disable and display dimmed links conditionally not applicable.

Table 4.3 Links to Avoid (or Links to 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.

Kinds of Links

- Within a page:
  - For long pages, include links to internal page content.

- Within a Web site:
  - On all pages include links to:
    - The Web site home page. [Home link-ready to start over]
    - Global Web site features. [Search facility]
    - Other main pages, navigation points, or categories.
    - The likely Web site starting point.
    - Main pages with links to the 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. [definition of terms and abbreviations]
    - New or changed content.
    - Web site Quit or Exit.
- 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. [To go outside of the Web site]
  - Provide an indication when a link goes outside the current site.

Link Maintenance

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

Maintaining a Sense of Place

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. [use of colours, patterns, graphics, font styles]

- Component presentation.
- Component organization and location.

Design Elements That Aid in Maintaining a Sense of Place

- Provide a home base. [a site's home page is a stable, concrete, and safe anchor point to escape to in times of difficulty]
- Use recurring navigation tools on all pages.
- Use recurring elements on all pages. [titles, banners, logos and icons]
- Provide page numbers for sequential pages.
- Provide ongoing feedback that shows where users are in a site. [breadcrumb trail]
- Provide on-demand aids that illustrate the user's location within a site.
  - Site maps.
  - Table of contents.
- Provide clearly written link labels. [indicate the function of the link, its destination, or its resulting action, reduce disorientation]

KINDS OF GRAPHICAL MENUS

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.

Design Guidelines for each of the Graphical Menus are as follows:

1. Menu Bar

- Proper usage:
  - To identify and provide access to common and frequently used application actions that takes 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. Menu bar is referred to as a collection of menu titles. A menu bar is the starting point for many dialogs. Menu bars often consist of a series of textual words, as represented in Figure 3.18

Figure 3.18: Menu bar composed of Text

Figure 3.19: Menu bar composed of buttons

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.

a. Display

- All primary windows must have a menu bar.
- All menu bars must have an associated pull-down menu containing at least two choices.
- Do not allow the user to turn off the display of the menu bar.
- If all the items in its associated pull-down menu are disabled, then disable the menu bar item.
    - Display the disabled item in a visually subdued manner.
    - However, the disabled pull-down menu must always be capable of being pulled down so that the choices may be seen.

b. Location

- Position choices horizontally over the entire row at the top of the screen, just below the screen title. [no more than about 7 or 8]
- A large number of choices may necessitate display over two rows.

c. Title

- The window title will be the menu bar title.

d. Item Descriptions

- The menu item descriptions must clearly reflect the kinds of choices available in the associated pull-down menus.
- 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. [if multiple-word item then include a hyphen between them
- Do not display choices that are never available to the user.

e. 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.

f. Layout

- Indent the first choice one space from the left margin.
- Leave at least three spaces between each of the succeeding choices (except for Help which will be right-justified).
- Leave one space between the final choice and the right margin.

g. Separation

- Separate the bar from the remainder of the screen by:
  - A different background, or
  - Solid lines above and below.

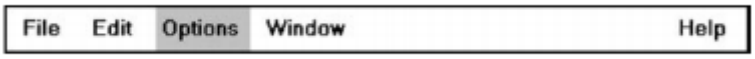
h. Other Components

- Keyboard equivalent mnemonics should be included on menu bars.
- Keyboard accelerators, to a window indicators, and cascade indicators need not be included.

i. Selection Indication

- Keyboard cursor:
  - Use a reverse video, or reverse , selection cursor to surround the choice.

- Cover the entire choice, including one blank space before and after the choice word.



- Pointer:
  - Use reverse video, or reverse, to highlight the selected choice. [black-white, blue-white, black-cyan]

2. 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.

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 5 to 10. Pull-downs are best suited for items represented textually, but graphical presentations, such as s, patterns, and shades, may also be used.

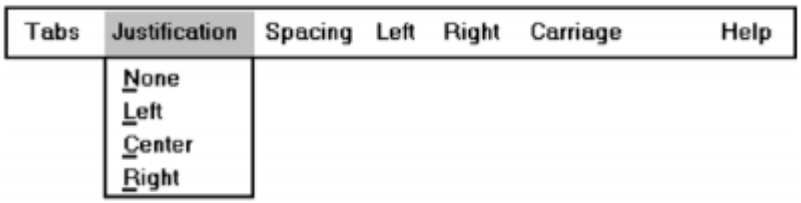


Figure 3.20: Menu bar pull-down

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.

- The 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.

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

b. Location
- Position the pull-down directly below the selected menu bar choice.

c. Size
- Must contain a minimum of two choices.
- Restrict to no more than 5 to 10 choices, preferably 8 or less.

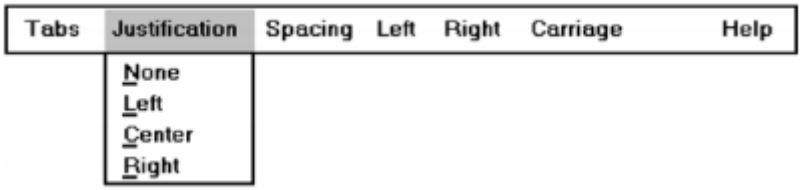
d. Title
- Not necessary on a pull-down menu. The title will be the name of the menu bar item chosen.

e. Item Descriptions
- Use mixed-case, headline-style words to describe choices.
    - If the choices can be displayed graphically, for example, as fill-in patterns, shades, or shades, or colours, textual descriptions are not necessary.
- Do not:
    - Identify a menu item by the same wording as its menu title.
    - Change the meaning of menu items through use of the Shift key.
    - Use scrolling in pull-downs.
    - Place instructions in pull-downs.

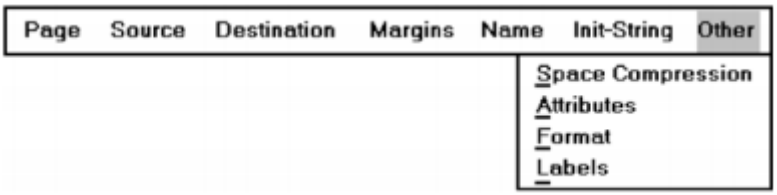
f. Organization
- Follow standard platform ordering schemes when they exist.
    - Place application-specific choices where they fit best.
- Place frequent or critical items at the top.
- Separate destructive choices from other choices.
- Align choices into columns, with:
    - Most frequent choices 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.

g. Layout
- Leave the menu bar choice leading to the pull-down highlighted in the selected manner (reverse video or reverse colour).

- Physically, the pull-down menu must be wide enough to accommodate the longest menu item description and its cascade or accelerator indicator.
- Align the first character of the pull-down descriptions under the second character of the applicable menu bar choice.
- Horizontally, separate the pull-down choice descriptions from the pull-down borders by two spaces on the left side and at least two spaces on the right side.
    - The left-side border will align with the left side of the highlighted menu bar choice.
    - The right-side border should extend, at minimum, to the right side of its highlighted menu bar choice.



    - Pull-downs for choices on the far right side of the menu bar, or long pull-down descriptions, may require alignment to the left of their menu bar choice to maintain visibility and clarity.



h. 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 colour as the choice descriptions.

i. Mark Toggles or Settings

- If a menu item establishes or changes the attributes of data or properties of the interface mark the pull-down choice or choices whose state is current or active “on”

- For nonexclusive items, display a check mark to the left of the item description. • If the two states of a setting are not obvious opposites, a pair of alternating menu item descriptions should be used to indicate the two states.
- For exclusive choices, precede the choice with a contrasting symbol such as a diamond or circle.

j. Pull-Downs Leading to Another Pull-Down

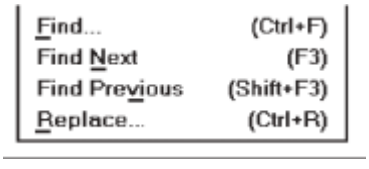
- If a pull-down choice leads to another pull-down, provide a cascade indicator as follows:
  - Place an arrow or right-pointing triangle after the choice description.
  - Align the triangles to the right side of the pull-down.
  - Display the triangle in the same colour as the choice descriptions.

k. Pull-Downs Leading to a Window

- For pull-down choices leading to a window:
  - Place an ellipsis (three dots) after the choice description.
  - Do not separate the dots from the description by a space.
  - Display the ellipsis in the same colour as the choice descriptions.

l. Keyboard Equivalents and Accelerators

- Provide unique mnemonic codes by which choices may be selected through the typewriter keyboard.
  - Indicate the mnemonic code by underlining the proper character.
- Provide key accelerators for choice selection.
  - Identify the keys by their actual key-top engravings.
  - Use a plus (+) sign to indicate that two or more keys must be pressed at the same time.
  - Enclose the key names within parentheses ( ).
  - Right-align the key names, beginning at least three spaces to the right of the longest choice description.
  - Display the key alternatives in the same colour as the choice descriptions.



m. Separation

- Separate the pull-down from the remainder of the screen, but visually relate it to the menu bar by:
  - Using a background colour the same as the menu bar.
  - Displaying choice descriptions in the same colour as the menu bar.
  - Incorporating a solid-line border completely around the pull-down in the same colour as the choice descriptions.
- A drop shadow (a heavier shaded line along two borders that meet) may also be included.

n. Selection Cursor

- Use a reverse video, or reverse colour, selection cursor the same colour as the menu bar to surround the choice.
- Create a consistently sized cursor as wide as the pull-down menu.

3. 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 3.21. Menu items that lead to cascading menus are typically indicated by a right-pointing triangle.

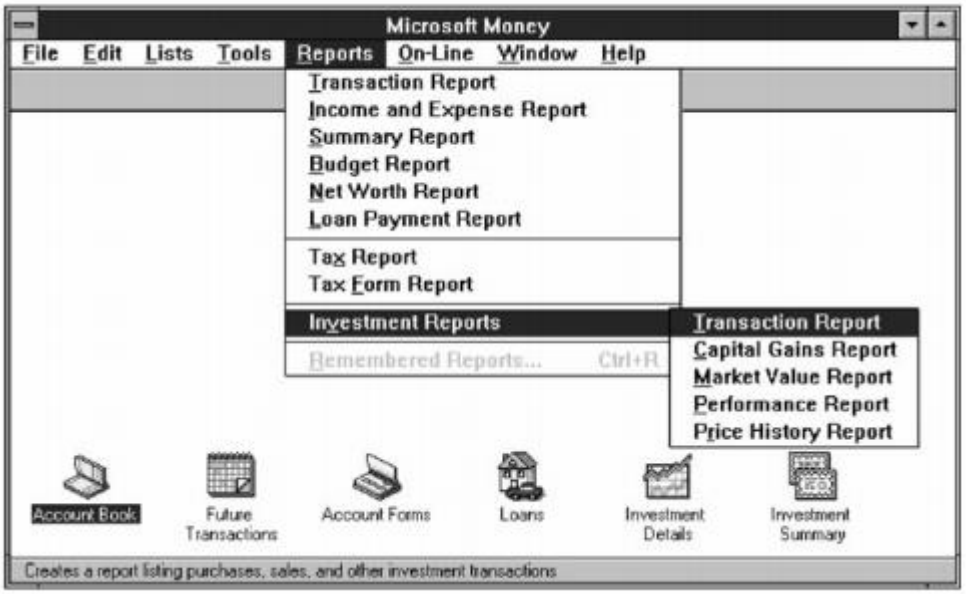


Figure-3.21: Cascading Menu

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.

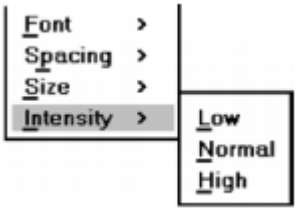
a. Cascade Indicator

- 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 colour as the choice descriptions.

b. Location

- Position the first choice in the cascading menu immediately to the right of the selected choice.

- Leave the choice leading to the cascading menu highlighted.



c. Levels

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

d. Title

- Not necessary on the cascading menu.
    - The title will be the name of the higher-level menu item chosen.

e. Other Guidelines

- Follow the organization, content, layout, separation, and selection cursor guidelines for the kind of menu from which the menu cascades.

4. 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 3.22

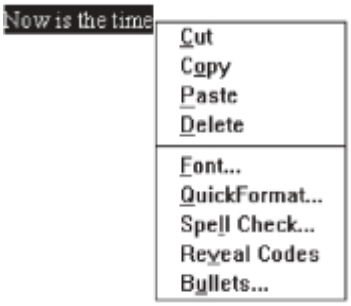


Figure 3.22: Pop-up Menu

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.

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.

a. 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.

b. 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.

c. Size

- Restrict the pop-up to no more than 5 to 10 choices, preferably 8 or less.

d. Title

- Not necessary on a pop-up menu.
- If included, clearly describe the menu's purpose.
- Locate in a centered position at the top.

- Display in uppercase or mixed-case letters.
- Separate it from the menu items by a line extending from the left menu border to the right border.

e. Other Guidelines

- Arrange logically organized and grouped choices into columns.
- If items are also contained in pull-down menus, organize pop-up menus in the same manner.
- Left-align choice descriptions.
- Use mixed-case headline-style words to describe choices.
- Separate groups with a solid line the length of the longest choice description.
- If the choice leads to a pop-up window, place an ellipsis after the choice description.
- To separate the pop-up from the screen background:
  - Use a contrasting, but complementary background.
  - Incorporate a solid line border around the pull-down.

5. 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

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.

6. 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.

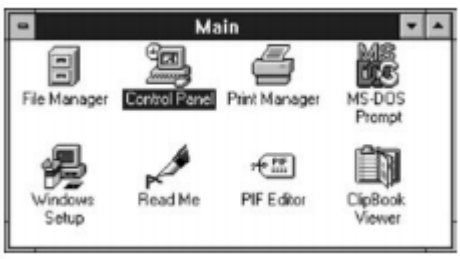


Figure 3.23: Iconic menu (From Microsoft Windows)

An iconic menu is the portrayal of menu items or objects in a graphic or pictorial form, as illustrated in Figure 3.23. 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.

7. Pie Menus

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

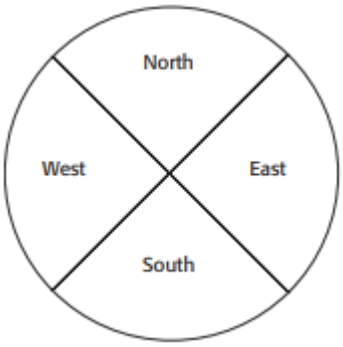


Figure 3.24: Pie Menu

A pie menu is a circular representation of menu items, as illustrated in Figure 3.24 that can be used as an alternative to a pull-down or pop-up menu. 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 edge is shorter. 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.

8. **Default Menu Items** Every system will provide a set of standard menu items. Using the default items will reduce design time and encourage interface consistency. System learning time will also be reduced

a. File

A standard element, the File menu provides all the commands needed to open, create, and save files. Some standard File functions are:

- New
- Open
- Close
- Save
- Save As
- Print Preview
- Print
- Exit

b. Edit

A standard element, the Edit menu provides commands that affect the state of selected objects. Some standard Edit functions are:

- Undo
- Cut
- Copy
- Paste
- Select All
- Find
- Replace

c. View

An optional element, the View menu provides commands that affect the perspective, details, and appearance of the application. They affect the view, not the data itself. The view functions are application-specific and include the following:

- Toolbars
- Status Bar
- Magnify
- Zoom In
- Zoom Out
- Grid Points

d. Window

The Window menu, an optional element, provides commands to manipulate entire windows. Included are items such as:

- New Window
- Arrange All
- Hide
- Show

e. Help

The Help menu, a standard element, provides Help commands, including:

- Contents
- Search for Help on
- How to Use Help
- About (Application)

Table 4.4 Menu Proper Usage Summary

Menu Bar	To identify and provide access to: • 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: • A small number of items (5–10). • 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 1–2 cascades.
Pop-Up Menu	For: • Frequent users. • Frequently used contextual commands. • A small number of items (5–10). • Items rarely changing in content. • Items that require a small amount of screen space.
Tear-Off Menu	For items: • Sometimes frequently selected. • Sometimes infrequently selected. • Small in number (5–10). • Rarely changing in content.
Iconic Menu	To designate applications available. To designate special functions within an application.

9. Functions Not Represented by Default Items

Having established the usability of the standard menu functions, additional system functions must be identified. Commands to accomplish these functions must be created and added to the pertinent menus. Command design guidelines include the following.

a. Labels

- General:
  - Provide a label for each command.
  - Use labels that indicate:
    - The purpose of the command, or
    - The result of what happens when the command is selected.
  - Use familiar, short, clear, concise words.
  - Use distinctive wording.
  - Use mixed case, with the first letter capitalized.
  - Begin commands with verbs or adjectives, not nouns.
  - Preferably, use only one word.
    - If multiple words are required for clarity, capitalize the first letter of each significant word.
    - Do not use sentences as labels.
  - Provide an ellipsis (. . .) to indicate that another window will result from selection of a command.
    - Do not use the ellipsis when the following window is a confirmation or warning.
- Dynamic labels:
  - As contexts change, dynamically change the label wording to make its meaning clearer in the new context.
    - For example, after a cut operation, Undo may be changed to Undo Cut.

b. Disabled Commands

- When a command is not available, indicate its disabled status by displaying it grayed out or subdued.
- If selection of a disabled command is attempted, provide a message in the information area that the ―Help‖ function will explain why it is disabled.

c. Navigation and Selection

- General
  - Permit multiple methods for selecting commands.
- Keyboard equivalents:
  - Assign a mnemonic for each command.
  - A mnemonic should be as meaningful as possible. Use:
    - The first letter of the command, or if duplications exist,
    - The first letter of another word in the command, or
    - Another significant consonant in the command.
  - For standard commands, use mnemonics provided by the tool set.
- Keyboard accelerators:
  - Assign keyboard accelerators for frequently used commands.
  - For standard commands, use keyboard accelerators provided by the tool set.

MODULE 4

CHAPTER 7. SELECT THE PROPER KIND OF WINDOWS

Syllabus: Windows – Characteristics, Components of a window, Window presentation styles, Types of window, Window management, organizing window functions, Window operations, Web systems, Characteristics of device-based controls

Objective

- To understand the purpose and usage of different kinds of windows for respective tasks.
- To identify proper input devices for implementing to the user based on their characteristics
- To identify proper screen-based controls for implementing to the user based on their characteristics

A window is an area of the screen, usually rectangular in shape, defined by a border that contains a particular view of some area of the computer or some portion of a person's dialog with the computer. It can be moved and rendered independently on the screen.

WINDOW CHARACTERISTICS

A window is seen to possess the following characteristics:

- A name or title, allowing it to be identified.
- A size in height and width (which can vary).
- A state, accessible or active, or not accessible. (Only active windows can have their contents altered.)
- Visibility—the portion that can be seen. (A window may be partially or fully hidden behind another window, or the information within a window may extend beyond the window's display area.)
- A location, relative to the display boundary.
- Presentation, that is, its arrangement in relation to other windows. It may be tiled, overlapping, or cascading.
- Management capabilities, methods for manipulation of the window on the screen.
- Its highlight, that is, the part that is selected.
- The function, task, or application to which it is dedicated.

The Attraction of Windows

While all the advantages and disadvantages of windows are still not completely understood, windows do seem to be useful in the following ways.

1. **Presentation of Different Levels of Information**: A document table of contents can be presented in a window. A chapter or topic selected from this window can be simultaneously displayed in more detail in an adjoining window.

2. **Presentation of Multiple Kinds of Information**: Variable information needed to complete a task can be displayed simultaneously in adjacent windows. For example in one window billing can be done and in one window stock. Maintenance can be done at the same time using windows. Significant windows could remain displayed so that details may be modified as needed prior
3. **Sequential Presentation of Levels or Kinds of Information**: Steps to accomplish a task can be sequentially presented through windows. Key windows may remain displayed, but others appear and disappear as necessary. This sequential preparation is especially useful if the information-collection process leads down various paths. The windows disappear after data entry, and additional windows appear when needed.
4. **Access to Different Sources of Information:** Independent sources of information may have to be accessed at the same time. This information may reside in different host computers, operating systems, applications, files, or areas of the same file. Example, a travel agent may have to compare several travel destinations for a particularly demanding client
5. **Combining Multiple Sources of Information:** Text from several documents may have to be reviewed and combined into one. Pertinent information is selected from one window and copied into another.
6. **Performing More Than One Task:** While waiting for a long, complex procedure to finish, another can be performed. Tasks of higher priority can interrupt less important ones and then the interrupted tasks can be preceded.
7. **Reminding:** It can be used to provide remainder through messages or popup or menus.
8. **Monitoring:** Data in one window can be modified and its effect on data in another window can be studied.
9. **Multiple Representations of the Same Task:** the same task can be represented in two different ways in two windows. For example a report can be given as table in one window and as a chart in another window.

Constraints in Window System Design

The problems with windowing systems can be attributed to three factors:

Historical considerations, Hardware limitations, and Human limitations

Historical considerations:

- Historically, system developers have been much more interested in solving hardware problems than in user considerations.
- This lack of guidelines makes it difficult to develop acceptable and agreeable window standards.
- The result is that developers of new systems create another new variation each time they design a product, and users must cope with a new interface each time they encounter a new windowing system.

Hardware Limitations

- Either seeing all the contents of one window is preferable to seeing small parts of many windows or the operational and visual complexity of multiple windows is not wanted.
- Poor screen resolution and graphics capability may also deter effective use of windows by not permitting sharp and realistic drawings and shapes.
- Also, the slower processing speeds and smaller memory sizes of some computers may inhibit the use of windows.

Human Limitations

- These window management operations are placed on top of other system operations, and window management can become an end in itself. This can severely detract from the task at hand.
- The results suggest that advantages for windows do exist, but they can be negated by excessive window manipulation requirements.
- Comparing full screens with screens containing overlapping windows, task completion times were longer with the window screens, but the non-window screens generated more user errors. After eliminating screen arrangement time, however, task solution times were shorter with windows
- It is also suggested that to be truly effective, window manipulation must occur implicitly as a result of user task actions, not as a result of explicit window management actions by the user.

Other Limitations

Other possible window problems include the necessity for window borders to consume valuable screen space, and that small windows providing access to large amounts of information can lead to excessive, bothersome scrolling

COMPONENTS OF A WINDOW

1. Frame
2. Title Bar
3. Title Bar Icon
4. Window Sizing Buttons
5. What's This? Button
6. Menu Bar
7. Status Bar
8. Scroll Bars
9. Split box
10. Tool Bar
11. Command Area

12. Size Grip
13. Work Area

Figure 8.1: Microsoft Windows Primary Window

Frame

A window will have a frame or border, usually rectangular in shape, to define its boundaries and distinguish it from other windows. While a border need not be rectangular, this shape is a preferred shape for most people.

Title Bar

The title bar is the top edge of the window, inside its border and extending its entire width. This title bar is also referred to by some platforms as the caption, caption bar, or title area. The title bar contains a descriptive title identifying the purpose or content of the window. The title bar also serves as a control point for moving the window and as an access point for commands that apply to a window. For example, as an access point, when a user clicks on the title bar using the secondary mouse button, the pop-up or shortcut menu for the window appears. Pressing the Alt-Spacebar key combination also displays the shortcut menu for the window. Title bars are included on all primary and secondary windows.

Title bar Icon

Located at the left corner of the title bar in a primary window, this button is used in Windows to retrieve a pull-down menu of commands that apply to the object in the window. It is 16X16 version of the icon of the object being viewed. Microsoft suggests that:

- If the window contains a tool or utility (that is, an application that does not create, load, and save its own data files), a small version of the application's icon should be placed there instead.
- If the application creates, loads, and saves documents or data files and the window represents the view of one of its files, a small version of the icon that represents its document or data file type should be placed there.
- Even if the user has not yet saved the file, display the data file icon rather than the application icon, and again display the data file icon after the user saves the file

Window Sizing Buttons

Located at the right corner of the title bar, these buttons are used to manipulate the size of a window. The leftmost button, the minimize button— inscribed with a short horizontal line toward the bottom of the button—is used to reduce a window to its minimum size, usually an icon. It also hides all associated windows. The maximize button—typically inscribed with a large box—enlarges a window to its maximum size, usually the entire screen. When a screen is maximized, the restore button replaces the maximize button, since the window can no longer be increased in size. A close button—typically inscribed with an X—closes the window.

When these buttons are displayed, use the following guidelines:

- When a window does not support a command, do not display its command button.
- The Close button always appears as the rightmost button. Leave a gap between it and any other buttons.
- The Minimize button always precedes the Maximize button.
- The Restore button always replaces the Maximize button or the Minimize button when that command is carried out.

What's This? Button

The What's This? Button, which appears on secondary windows and dialog boxes, is used to invoke the What's This? Windows command to provide contextual Help about objects displayed within a secondary window. On a primary window this command is accessed from the Help drop-down menu. This command may also be included as a button on a toolbar or as a command on a pop-up menu for a specific object.

Figure 8.2: What's This? Button

Menu Bar

A menu bar is used to organize and provide access to actions. It is located horizontally at the top of the window, just below the title bar. A menu bar contains a list of topics or items that, when selected, are displayed on a pull-down menu beneath the choice.

Status Bar

Information of use to the user can be displayed in a designated screen area or areas. They may be located at the top of the screen in some platforms and called a status area, or at the screen's bottom. Microsoft recommends the bottom location and refers to this area as the status bar. It is also referred to by other platforms as a message area or message bar. The status bar to display information about the current state of what is being viewed in the window, descriptive messages about a selected menu or toolbar button, or other non-interactive information.

Scroll Bars

When all display information cannot be presented in a window, the additional information must be found and made visible. This is accomplished by scrolling the display's contents through use of a scroll bar. A scroll bar is an elongated rectangular container consisting of a scroll area or shaft, a slider box or elevator, and arrows or anchors at each end. For vertical scrolling, the scroll bar is positioned at the far right side of the work

Split Box

A window can be split into two or more pieces or panes by manipulating a split box located above a vertical scroll bar or to the left of a horizontal scroll bar. A split box is sometimes referred to as a split bar. A window can be split into two or more separate viewing areas that are called panes. A split window allows the user to:

- Examine two parts of a document at the same time.
- Display different, yet simultaneous, views of the same information.

Toolbar

Toolbars are permanently displayed panels or arrays of choices or commands that must be accessed quickly. They are sometimes called command bars. Toolbars are designed to provide quick access to specific commands or options. Specialized toolbars are sometimes referred to as ribbons, toolboxes, rulers, or palettes.

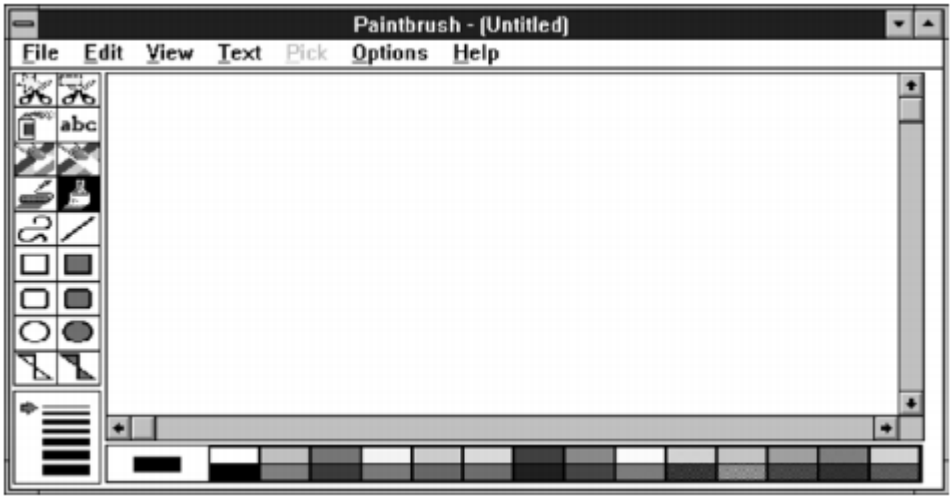


Figure 8.3: Toolbars

Command Area

In situations where it is useful for a command to be typed into a screen, a command area can be provided. The desired location of the command area is at the bottom of the window.

Size Grip

A size grip is a Microsoft Windows special handle included in a window to permit it to be resized. When the grip is dragged the window resizes, following the same conventions as the sizing border. Three angled parallel lines in the lower-right corner of a window designate the size grip. If the window possesses a status bar, the grip is positioned at the bar's right end. Otherwise, it is located at the bottom of a vertical scroll bar, the right side of a horizontal scroll bar, or the junction point of the two bars.

Work Area

The work area is the portion of the screen where the user performs tasks. It is the open area inside the window's border and contains relevant peripheral screen components such as the menu bar, scroll bars, or message bars. The work area may also be referred to as the client area.

Table 5.1 Microsoft Windows Components

COMPONENT	WINDOWS CONTAINING COMPONENT		
	PRIMARY	SECONDARY	DIALOG BOX
Frame or Border • Boundary to define shape. • If sizable, contains control points for resizing.	X	X	X
Title Bar Text • Name of object being viewed in window. • Control point for moving window.	X	X	X
Title Bar Icon • Small version of icon for object being viewed. • Access point for commands that apply to the object.	X		
Title Bar Buttons • Shortcuts to specific commands.	X	X	X
Close	X	X	X
Minimize/Maximize/Restore	X		
What's This? – Displays context-sensitive Help about any object displayed on window.		X	X
Menu Bar • Provides basic and common application commands.	X		
Status Bar • An area used to display status information about what is displayed in window.	X		
Scroll Bar • Standard control to support scrolling.	X		
Size Grip • Control to resize window, located at right side of status bar.	X		

WINDOW PRESENTATION STYLES

The presentation style of a window refers to its spatial relationship to other windows. There are two basic styles, commonly called tiled or overlapping.

Tiled Windows

Tiled windows derive their name from common floor or wall tile. Tiled windows appear in one plane on the screen and expand or contract to fill up the display surface, as needed. Most systems provide two-dimensional tiled windows, adjustable in both height and width.

Figure 8.4: Tiled windows

Advantages:

- The system usually allocates and positions windows for the user, eliminating the necessity to make positioning decisions.
- Open windows are always visible, eliminating the possibility of them being lost and forgotten.
- Every window is always completely visible, eliminating the possibility of information being hidden.
- They are perceived as fewer complexes than overlapping windows, possibly because there are fewer management operations or they seem less ―magical.
- They are easier, according to studies, for novice or inexperienced people to learn and use.
- They yield better user performance for tasks where the data requires little window manipulation to complete the task.

Disadvantages

- Only a limited number can be displayed in the screen area available.
- As windows are opened or closed, existing windows change in size. This can be annoying.
- As windows change in size or position, the movement can be disconcerting.
- As the number of displayed windows increases, each window can get very tiny.
- The changes in sizes and locations made by the system are difficult to predict.
- The configuration of windows provided by the system may not meet the user's needs.
- They are perceived as crowded and more visually complex because window borders are flush against one another, and they fill up the whole screen. Crowding is accentuated if borders contain scroll bars or control icons. Viewer attention may be drawn to the border, not the data.
- They permit less user control because the system actively manages the windows.

Overlapping Windows

Overlapping windows may be placed on top of one another like papers on a desk. They possess a three-dimensional quality, appearing to lie on different planes.

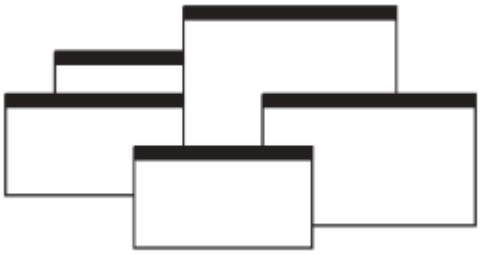


Figure 8.5: Overlapping windows

Advantages:

- Visually, their look is three-dimensional, resembling the desktop that is familiar to the user.
- Greater control allows the user to organize the windows to meet his or her needs.
- Windows can maintain larger sizes.
- Windows can maintain consistent sizes.
- Windows can maintain consistent positions.
- Screen space conservation is not a problem, because windows can be placed on top of one another.
- There is less pressure to close or delete windows no longer needed.
- The possibility exists for less visual crowding and complexity. Larger borders can be maintained around window information, and the window is more clearly set off against its background. Windows can also be expanded to fill the entire display.
- They yield better user performance for tasks where the data requires much window manipulation to complete the task.

Disadvantages

- They are operationally much more complex than tiled windows. More control functions require greater user attention and manipulation.
- Information in windows can be obscured behind other windows.
- Windows themselves can be lost behind other windows and be presumed not to exist.
- That overlapping windows represent a three-dimensional space is not always realized by the user.
- Control freedom increases the possibility for greater visual complexity and crowding. Too many windows, or improper offsetting, can be visually overwhelming.

Cascading Windows

A special type of overlapping window has the windows automatically arranged in a regular progression. Each window is slightly offset from others, as illustrated in Figure 8.6

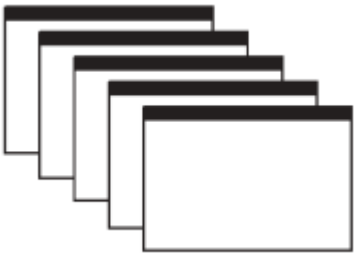


Figure 8.6: Cascading windows

Advantages

- No window is ever completely hidden.
- Bringing any window to the front is easier.
- It provides simplicity in visual presentation and cleanness.

Picking a Presentation Style

- Use tiled windows for:
  - Single-task activities.
  - Data that needs to be seen simultaneously.
  - Tasks requiring little window manipulation.
  - Novice or inexperienced users.
- Use overlapping windows for:
  - Switching between tasks.
  - Tasks necessitating a greater amount of window manipulation.
  - Expert or experienced users.
  - Unpredictable display contents.

TYPES OF WINDOWS

People's tasks must be structured into a series of windows. The type of window used will depend on the nature and flow of the task.

1. Primary window
2. Secondary window
3. Dialog boxes
4. Property sheets and Property Inspectors
5. Message boxes
6. Palette windows
7. Pop-up windows

Primary Window

Proper usage:

- Should represent an independent function or application.
- Use to present constantly used window components and controls.
    - Menu bar items that are:
        — Used frequently.
        — Used by most, or all, primary or secondary windows.
    - Controls used by dependent windows.
- Use for presenting information that is continually updated.
    - For example, date and time.
- Use for providing context for dependent windows to be created.
- Do not:
    - Divide an independent function into two or more primary windows.
    - Present unrelated functions in one primary window.

Figure 8.7: Microsoft Windows primary window

The primary window is the first one that appears on a screen when an activity or action is started. It is required for every function or application, possessing a menu bar and some basic action controls. It has also been variously referred to as the application window or the main window. In addition, it may be referred to as the parent window if one or more child windows exist. The primary window is the main focal point of the user's activities and should represent an independent function

Secondary Windows

- Proper usage:
    - For performing subordinate, supplemental, or ancillary actions that are:
        - Extended or more complex in nature.
        - Related to objects in the primary window.
    - For presenting frequently or occasionally used window components.
- Important guidelines:
    - Should typically not appear as an entry on the taskbar.
    - A secondary window should not be larger than 263 dialog units x 263 dialog units.

Figure 8.8: Microsoft Windows secondary window

Secondary windows may be dependent upon a primary window or displayed independently of the primary window. They structurally resemble a primary window, possessing some of the same action controls (Close button) and possibly a What's This? button.

A dependent secondary window is one common type. It can only be displayed from a command on the interface of its primary window. It is typically associated with a single data object, and appears on top of the active window when requested. It is movable, and scrollable.

An independent secondary window can be opened independently of a primary window—for example, a property sheet displayed when the user clicks the Properties command on the menu of a desktop icon.

Microsoft recommends not displaying any secondary window larger than 263 dialog units × 263 dialog units. Microsoft defines size and location of user-interface elements not in pixels but in dialog units (DLUs), a device-independent unit of measure.

- One horizontal DLU is equal to one-fourth of the average character width for the current system font.
- One vertical DLU is equal to one-eighth of the average character height for the current system font

A secondary window can be of two types modal or modeless

Modal:

- Use when interaction with any other window must not be permitted.
- Use for:
  - Presenting information. — For example, messages (sometimes called a message box).
  - Receiving user input. — For example, data or information (sometimes called a prompt box).
  - Asking questions. — For example, data, information, or directions (sometimes called a question box).
  - Use carefully because it constrains what the user can do.

Modeless:

- Use when interaction with other windows must be permitted.
- Use when interaction with other windows must be repeated.

Multiple secondary windows needed to complete a task may be presented in two forms, cascading or expanding

Cascading:

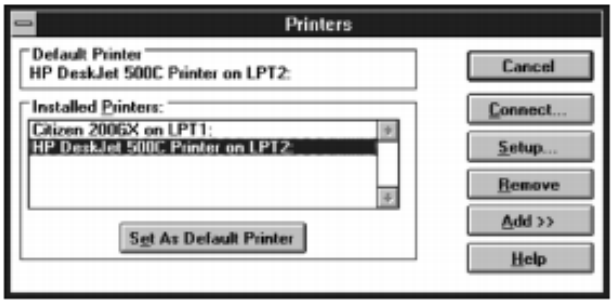


Figure 5.9 Printers secondary window with Connect... cascade button.

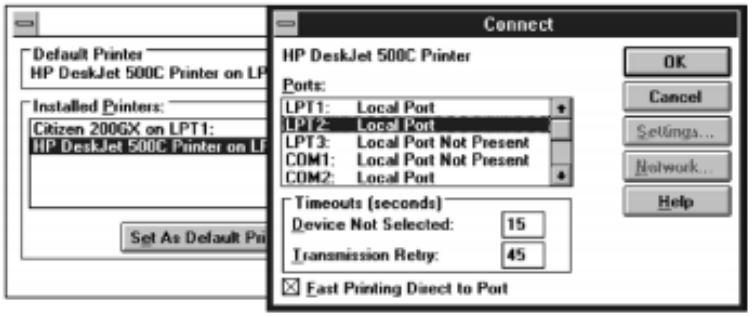


Figure 5.10 Cascading Connect secondary window.

- Purpose:
  - To provide advanced options at a lower level in a complex dialog.
- Guidelines:
  - Provide a command button leading to the next dialog box with a “To a Window” indicator, an ellipsis (. . .).
  - Present the additional dialog box in cascaded form.
  - Provide no more than two cascades in a given path.
  - Do not cover previous critical information.
    — Title Bar.
    — Relevant displayed information.
  - If independent, close the secondary window from which it was opened.

Unfolding:

- Purpose:

  - To provide advanced options at the same level in a complex dialog.
- Guidelines:
  - Provide a command button with an expanding dialog symbol (>>).
  - Expand to right or downward.

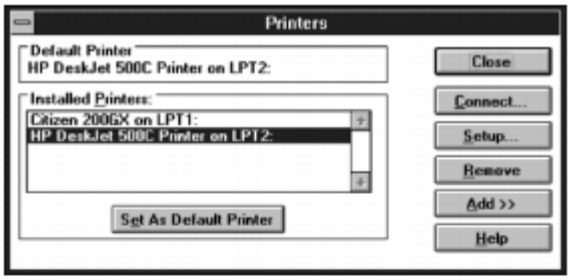


Figure 5.11 Printers secondary window with *Add* >> unfolding button.

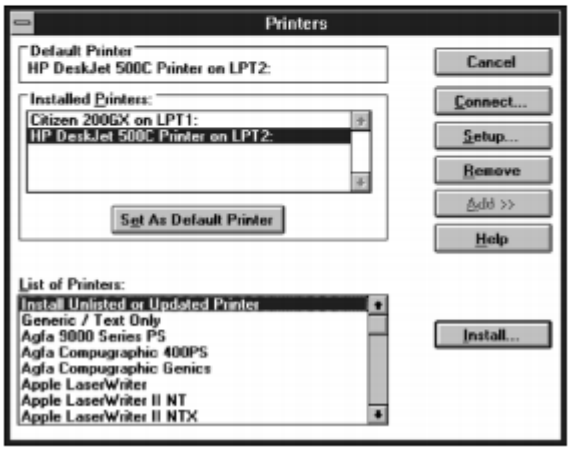


Figure 5.12 Unfolded Printers secondary window.

Dialog Boxes

- Use for presenting brief messages.
- Use for requesting specific, transient actions.
- Use for performing actions that:
  — Take a short time to complete.
  — Are not frequently changed.
- Command buttons to include:
  — OK.
  — Cancel.
  — Others as necessary.

Dialog boxes are always displayed from another window, either primary or secondary, or another dialog box. They may appear as a result of a command button being activated or a menu choice being selected, or they may be presented automatically by the system when a condition exists that requires the user's attention or additional input.

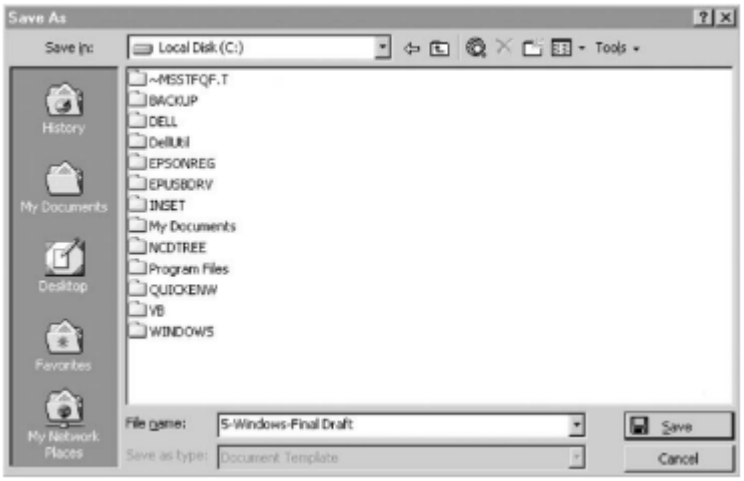


Figure 8.9: Microsoft Windows dialog box

Most windowing systems provide standard dialog boxes for common functions, some examples being Open, Save As, and Print. Many platforms also recommend a set of standard command buttons for use in the various kinds of dialog boxes, such as OK, Cancel, and so on. Dialog boxes are of two types, modal and modeless, as recently described. They may also cascade or unfold.

Secondary windows provide two other techniques for displaying properties, property sheets and property inspectors.

Property Sheets

- Use for presenting the complete set of properties for an object.
- Categorize and group within property pages, as necessary.
    - Use tabbed property pages for grouping peer-related property sets.
    - The recommended sizes for property sheets are:
        - 252 DLUs wide x 218 DLUs high
        - 227 DLUs wide x 215 DLUs high
        - 212 DLUs wide x 188 DLUs high
    - Command buttons to include:
        - OK.
        - Cancel.
        - Apply.
        - Reset.
        - Others as necessary.
    - For single property sheets, place the commands on the sheet.
    - For tabbed property pages, place the commands outside the tabbed pages.

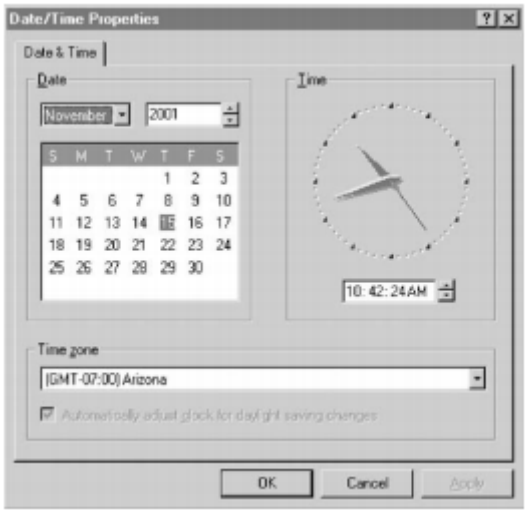


Figure 8.10: Microsoft Windows property sheet

Figure 8.11: Microsoft Windows property sheet tabbed pages

Property Inspectors

- Use for displaying only the most common or frequently accessed objects properties.
- Make changes dynamically.

Figure 8.12: Microsoft Windows property inspector

Property inspectors and property sheets are not exclusive interfaces. Both can be included in an interface. The most common or frequently accessed properties can be displayed in a property inspector and the complete set in the property sheet.

Message Boxes

- Use for displaying a message about a particular situation or condition.
- Command buttons to include:
  — OK.
  — Cancel.
  — Help.
  — Yes and No.
  — Stop.
  — Buttons to correct the action that caused the message box to be displayed. [save and delete]
- Enable the title bar close box only if the message includes a cancel button.
- Designate the most frequent or least destructive option as the default command

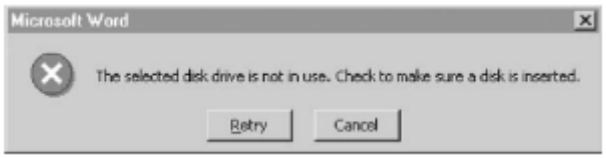


Figure 8.13: Microsoft Windows message box

Palette Windows

- Use to present a set of controls.
- Design as resizable.
  — Alternately, design them as fixed in size.

Figure 8.14: Microsoft Windows palette window

Palette windows are modeless secondary windows that present a set of controls, as shown in Figure 8.13. Palette windows are distinguished by their visual appearance, a collection of images, colours or patterns. The title bar for a palette window is shorter and includes only a close button.

Pop-up Windows

- Use pop-up windows to display:

— Additional information when an abbreviated form of the information is the main presentation.
— Textual labels for graphical controls. [ToolTips and balloon tips]
— Context-sensitive Help information

Figure 8.15: Microsoft Windows pop-up window

Table 5.2 Microsoft Windows Window Types and Components

PRIMARY WINDOW	
Purpose:	To perform a major interaction.
Components:	Frame or border. Title bar. –Access point for commands that apply to the window, with commands displayed in a pop-up menu. Title Bar icon. –Small version of the icon of the object being viewed. –Access point for commands that apply to the object being displayed in the window, with commands displayed in a pop-up window. Title bar text. Title bar buttons to: close/minimize/maximize /restore a window. Menu bar. Status bar. Scroll bar. Size grip.
SECONDARY WINDOWS	
Purpose:	To obtain or display supplemental information related to the objects in the primary window.
Components:	Frame or border. Title bar. Title bar text. Close button. What's This? button. –Context-sensitive Help about components displayed in the window; this is optional.
Kinds:	Modal and modeless.

(continues)

WINDOW MANAGEMENT

Microsoft Windows also provides several window management schemes, a single document interface, a multiple-document interface, workbooks, and projects.

Single-Document Interface

- Description:
  — A single primary window with a set of secondary windows.
- Proper usage:
  — Where object and window have a simple, one-to-one relationship.
  — Where the object's primary presentation or use is as a single unit.
  — To support alternate views with a control that allows the view to be changed.
  — To support simultaneous views by splitting the window into panes.

Table 5.2 Continued

SECONDARY WINDOWS	
Dialog Boxes	
Purpose:	To obtain additional information needed to carry out a particular command or task.
Description:	Secondary window. Contains the following common dialog box interfaces: – Open/Replace/Find. – Save As /Print/Print Setup. – Page Setup/Font/Color.
Property Inspectors	
Purpose:	To display the most common or frequently accessed properties of a current selection, usually of a particular type of object.
Description:	A modeless secondary window. Typically modal with respect to the object for which it displays properties.
Usage:	Displayed when requested from selected object.
Property Sheets	
Purpose:	For presenting the complete set of properties for an object.
Description:	A modeless secondary window. Typically modal with respect to the object for which it displays properties.
Usage:	Displayed when requested from selected object.
Message Boxes	
Purpose:	To provide information about a particular situation or condition.
Description:	Secondary window. Types of message boxes: – Information/Warning/Critical.
Palette Windows	
Purpose:	To present a set of controls such as palettes or toolbars.
Description:	Modeless secondary window.

Table 5.2 Continued

SECONDARY WINDOWS	
Pop-Up Windows	
Purpose:	To display additional information when an abbreviated form of the information is the main presentation.
Description:	Secondary window. Does not contain standard secondary window components such as title bar and close button. Example: ToolTip.

Advantages:

- Most common usage.
- Window manipulation is easier and less confusing.
- Data-centered approach.

Disadvantage:

- Information is displayed or edited in separate windows.

Multiple-Document Interface

- Description:
  - A technique for managing a set of windows where documents are opened into windows.
  - Contains:
    - A single primary window, called the parent.
    - A set of related document or child windows, each also essentially a primary window.
  - Each child window is constrained to appear only within the parent window.
  - The child windows share the parent window's operational elements.
  - The parent window's elements can be dynamically changed to reflect the requirements of the active child window.
- Proper usage:
  - To present multiple occurrences of an object.
  - To compare data within two or more windows.
  - To present multiple parts of an application.
  - Best suited for viewing homogeneous object types.
  - To clearly segregate the objects and their windows used in a task.
- Advantages:
  - The child windows share the parent window's interface components (menus, toolbars, and status bars), making it a very space-efficient interface.
  - Useful for managing a set of objects.

  - Provides a grouping and focus for a set of activities within the larger environment of the desktop.
- Disadvantages:
  - Reinforces an application as the primary focus.
  - Containment for secondary windows within child windows does not exist, obscuring window relationships and possibly creating confusion.
  - Because the parent window does not actually contain objects, context cannot always be maintained on closing and opening.
  - The relationship between files and their windows is abstract, making an MDI application more challenging for beginning users to learn.
  - Confining child windows to the parent window can be inconvenient or inappropriate for some tasks.
  - The nested nature of child windows may make it difficult for the user to distinguish a child window in a parent window from a primary window that is a peer with the parent window but is positioned on top.

Workbooks

- Description:
  - A window or task management technique that consists of a set of views organized like a tabbed notebook.
  - It is based upon the metaphor of a book or notebook.
  - Views of objects are presented as sections within the workbook's primary windows; child windows do not exist.
  - Each section represents a view of data.
  - Tabs can be included and used to navigate between sections.
  - Otherwise, its characteristics and behaviour are similar to those of the multiple document interface with all child windows maximized.
- Proper usage:
  - To manage a set of views of an object.
  - To optimize quick navigation of multiple views.
  - For content where the order of the sections is significant.
- Advantages:
  - Provides a grouping and focus for a set of activities within the larger environment of the desktop.
  - Conserves screen real estate.
  - Provides the greater simplicity of the single-document window interface.
  - Provides greater simplicity by eliminating child window management.
  - Preserves some management capabilities of the multiple-document interface.
- Disadvantage:
  - Cannot present simultaneous views.

Projects

- Description:
  - A technique that consists of a container: a project window holding a set of objects.
  - The objects being held within the project window can be opened in primary windows that are peers with the project window.
  - Visual containment of the peer windows within the project window is not necessary.
  - Each opened peer window must possess its own menu bar and other interface elements.
  - Each opened peer window can have its own entry on the task bar.
  - When a project window is closed, all the peer windows of objects also close.
  - When the project window is opened, the peer windows of the contained objects are restored to their former positions.
  - Peer windows of a project may be restored without the project window itself being restored.
- Proper usage:
  - To manage a set of objects that do not necessarily need to be contained.
  - When child windows are not to be constrained.
- Advantages:
  - Provides a grouping and focus for a set of activities within the larger environment of the desktop.
  - Preserves some management capabilities of the multiple document interface.
  - Provides the greatest flexibility in the placement and arrangement of windows.
- Disadvantage:
  - Increased complexity due to difficulty in differentiating peer primary windows of the project from windows of other applications.

ORGANIZING WINDOW FUNCTIONS

Window Organization

- Organize windows to support user tasks.
- Support the most common tasks in the most efficient sequence of steps.
- Use primary windows to:
  — Begin an interaction and provide a top-level context for dependent windows.
  — Perform a major interaction.
- Use secondary windows to:
  — Extend the interaction.
  — Obtain or display supplemental information related to the primary window.

- Use dialog boxes for:
  — Infrequently used or needed information.
  —Nice-to-know information.

Number of Windows

- Minimize the number of windows needed to accomplish an objective.

The general rule:

- Minimize the number of windows used to accomplish an objective.
- Use a single window whenever possible. Consider, however, the user's task.
- Don't clutter up a single window with rarely used information when it can be placed on a second, infrequently used, window.

WINDOW OPERATIONS

The following are the operational guidelines for Window operations

Active Window

- A window should be made active with as few steps as possible.
- Visually differentiate the active window from other windows.

General Guidelines

- Design easy to use and learn windowing operations.
  — Direct manipulation seems to be a faster and more intuitive interaction style than indirect manipulation for many windowing operations.
- Minimize the number of window operations necessary to achieve a desired effect.
- Make navigating between windows particularly easy and efficient to do.
- Make the setting up of windows particularly easy to remember.
- In overlapping systems, provide powerful commands for arranging windows on the screen in user-tailorable configurations.

Opening a Window

- Provide an iconic representation or textual list of available windows.
  — If opening with an expansion of an icon, animate the icon expansion.
- When opening a window:
  — Position the opening window in the most forward plane of the screen.
  — Adapt the window to the size and shape of the monitor on which it will be presented.
  — Designate it as the active window.
  — Set it off against a neutral background.
  — Ensure that its title bar is visible.
- When a primary window is opened or restored, position it on top.

— Restore all secondary windows to the states that existed when the primary window was closed.

- When a dependent secondary window is opened, position it on top of its associated primary window.
  — Position a secondary window with peer windows on top of its peers.
  — Present layered or cascaded windows with any related peer secondary windows.
- When a dependent secondary window is activated, its primary window and related peer windows should also be positioned at the top.
- If more than one object is selected and opened, display each object in a separate window. Designate the last window selected as the active window.
- Display a window in the same state as when it was last accessed.
  — If the task, however, requires a particular sequence of windows, use a fixed or consistent presentation sequence.
- With tiled windows, provide an easy way to resize and move newly opened windows.

Sizing Windows

- Provide large-enough windows to:
  — Present all relevant and expected information for the task.
  — Avoid hiding important information.
  — Avoid crowding or visual confusion.
  — Minimize the need for scrolling.
    • But use less than the full size of the entire screen.
- If a window is too large, determine:
  — Is all the information needed?
  — Is all the information related?
- Otherwise, make the window as small as possible.
  — Optimum window sizes:
    • For text, about 12 lines.
    • For alphanumeric information, about seven lines.
- Larger windows seem to have these advantages:
  - They permit displaying of more information.
  - They facilitate learning: Data relationships and groupings are more obvious.
  - Less window manipulation requirements exist.
  - Breadth is preferred to depth (based on menu research).
  - More efficient data validation and data correction can be performed.
- Disadvantages include:
  - Longer pointer movements are required.
  - Windows are more crowded.
  - More visual scanning is required.
  - Other windows more easily obscure parts of the window.

- It is not as easy to hide inappropriate data.

Window Placement

- Considerations:
  — In placing a window on the display, consider:
    • The use of the window.
    • The overall display dimensions.
    • The reason for the window's appearance.
- General:
  — Position the window so it is entirely visible.
  — If the window is being restored, place the window where it last appeared.
  — If the window is new, and a location has not yet been established, place it:
    • At the point of the viewer's attention, usually the location of the pointer or cursor.
    • In a position convenient to navigate to.
    • So that it is not obscuring important or related underlying window information.
  — For multiple windows, give each additional window its own unique and discernible location.
    • A cascading presentation is recommended.
  — In a multiple-monitor configuration, display the secondary window on the same monitor as its primary window.
  — If none of the above location considerations apply, then:
    • Horizontally center a secondary window within its primary window just below the title bar, menu bar, and any docked toolbars.
  — If the user then moves the window, display it at this new location the next time the user opens the window.
    • Adjust it as necessary to the current display configuration.
  — Do not let the user move a window to a position where it cannot be easily repositioned.
- Dialog boxes:
  — If the dialog box relates to the entire system, center it on screen.
  — Keep key information on the underlying screen visible.
  — If one dialog box calls another, make the new one movable whenever possible.

Window Separation

Crisply, clearly, and pleasingly demarcate a window from the background of the screen on which it appears.

- Provide a surrounding solid line border for the window.
- Provide a window background that sets the window off well against the overall screen background.
- Consider incorporating a drop shadow beneath the window.

Moving a Window

- Permit the user to change the position of all windows.
- Change the pointer shape to indicate that the move selection is successful.
- Move the entire window as the pointer moves.
  — If it is impossible to move the entire window, move the window outline while leaving the window displayed in its original position.
  Permit the moving of a window without its being active.

Resizing a Window

- Permit the user to change the size of primary windows. — Unless the information displayed in the window is fixed or cannot be scaled to provide more information.
- Change the pointer shape to indicate that the resizing selection is successful.
- The simplest operation is to anchor the upper-left corner and resize from the lower right corner.
  — Also permit resizing from any point on the window.
- Show the changing window as the pointer moves.
  — If it is impossible to show the entire window being resized, show the window's outline while leaving the window displayed in its original position.
- When window size changes and content remains the same:
  — Change image size proportionally as window size changes.
- If resizing creates a window or image too small for easy use, do one of the following:
  — Clip (truncate) information arranged in some logical structure or layout when minimum size is attained, or
  — When no layout considerations exist, format (restructure) information as size is reduced, or
  — Remove less useful information (if it can be determined), or
  — When minimum size is attained, replace information with a message that indicates that the minimum size has been reached and that the window must be enlarged to continue working.
- Permit resizing a window without its being active.

Other Operations

Permit primary windows to be maximized, minimized, and restored.

Window Shuffling

- Window shuffling must be easy to accomplish.

Keyboard Control/Mouse less Operation

- Window actions should be capable of being performed through the keyboard as well as with a mouse.
- Keyboard alternatives should be designated through use of mnemonic codes as much as possible.
- Keyboard designations should be capable of being modified by the user.

Closing a Window

- Close a window when:
  — The user requests that it be closed.
  — The user performs the action required in the window.
  — The window has no further relevance.
- If a primary window is closed, also close all of its secondary windows.
- When a window is closed, save its current state, including size and position, for use when the window is opened again.

WEB SYSTEMS

Web systems have limited windowing capabilities. The frame concept does provide window-like ability, and JavaScript does provide pop-up windows.

Frames

- Description:
  — Multiple Web screen panes that permit the displaying of multiple documents on a page.
  — These documents can be independently viewed, scrolled, and updated.
  — The documents are presented in a tiled format.
- Proper usage:
  — For content expected to change frequently.
  — To allow users to change partial screen content.
  — To permit users to compare multiple pieces of information.
- Guidelines:
  — Use only a few frames (three or less) at a given time.
  — Choose sizes based upon the type of information to be presented.
  — Never force viewers to resize frames to see information.
  — Never use more than one scrolling region on a page.
- Advantages
  - They decrease the user's need to jump back and forth between screens, thereby reducing navigation-related cognitive overhead.
  - They increase the user's opportunity to request, view, and compare multiple sources of information.

- They allow content pages to be developed independently of navigation pages.

- Disadvantages
  - The difference between a single Web page and a page with frames is not always obvious to the user.
  - They suffer some of the shortcomings of tiled screens:
    — Only a limited number can be displayed in the available screen area.
    — They are perceived as crowded and more visually complex because frame borders are flush against one another and they fill up the whole screen. Crowding is accentuated if the borders contain scroll bars and/or control icons. Viewer attention may be drawn to the border, not the data.
  - Frames-based pages behave differently from regular Web pages.
    — Page-printing difficulties and problems can exist.
    — Page interaction can be clumsy.
    — URLs cannot be e-mailed to other users.
  - Frames will not work on older browsers.

Pop-Up Windows

- Be extremely cautious in the use of pop-up windows.

CHAPTER 8

SELECT THE PROPER DEVICE-BASED CONTROLS

Device-based controls, often called input devices, are the mechanisms through which people communicate their desires to the system.

CHARACTERISTICS OF DEVICE-BASED CONTROLS

Several specific tasks are performed using graphical systems.

- To point at an object on the screen.
- To select the object or identify it as the focus of attention.
- To drag an object across the screen.
- To draw something free form on the screen.
- To track or follow a moving object.
- To orient or position an object.
- To enter or manipulate data or information.

Direct and Indirect Devices

- Direct devices are operated on the screen itself. Examples include the light pen, the finger, and voice.
- Indirect devices are operated in a location other than the screen, most often on the desktop.

TRACKBALL

- Description:
  - A spherical object (ball) that rotates freely in all directions in its socket.
  - Direction and speed is tracked and translated into cursor movement.
- Advantages:
  - Direct relationship between hand and pointer movement in terms of direction and speed.
  - Does not obscure vision of screen.
  - Does not require additional desk space (if mounted on keyboard).
- Disadvantages:
  - Movement is indirect, in a plane different from the screen.
  - No direct relationship exists between hand and pointer movement in terms of distance.
  - Requires a degree of eye-hand coordination.
  - Requires hand to be removed from keyboard keys.
  - Requires different hand movements.
  - Requires hand to be removed from keyboard (if not mounted on keyboard).
  - Requires additional desk space (if not mounted on keyboard).

- May be difficult to control.
- May be fatiguing to use over extended time.

JOYSTICK

- Description:
  - A stick or bat-shaped device anchored at the bottom.
  - Variable in size, smaller ones being operated by fingers, larger ones requiring the whole hand.
  - Variable in cursor direction movement method, force joysticks respond to pressure, movable ones respond to movement.
  - Variable in degree of movement allowed, from horizontal-vertical only to continuous.
- Advantages:
  - Direct relationship between hand and pointer movement in terms of direction.
  - Does not obscure vision of screen.
  - Does not require additional desk space (if mounted on keyboard).
- Disadvantages:
  - Movement indirect, in plane different from screen.
  - Indirect relationship between hand and pointer in terms of speed and distance.
  - Requires a degree of eye-hand coordination.
  - Requires hand to be removed from keyboard keys.
  - Requires different hand movements to use.
  - Requires hand to be removed from keyboard (if not mounted on keyboard).
  - Requires additional desk space (if not mounted on keyboard).
  - May be fatiguing to use over extended time.
  - May be slow and inaccurate.

GRAPHIC TABLET

- Description:
  - Pressure-, heat-, light-, or light-blockage-sensitive horizontal surfaces that lie on the desktop or keyboard.
  - May be operated with fingers, light pen, or objects like a stylus or pencil.
  - Pointer imitates movements on tablet.
- Advantages:
  - Direct relationship between touch movements and pointer movements in terms of direction, distance, and speed.
  - More comfortable horizontal operating plane.
  - Does not obscure vision of screen.
- Disadvantages:

- Movement is indirect, in a plane different from screen.
- Requires hand to be removed from keyboard.
- Requires hand to be removed from keyboard keys.
- Requires different hand movements to use.
- Requires additional desk space.
- Finger may be too large for accuracy with small objects

TOUCH SCREEN

- Description:
  - A special surface on the screen sensitive to finger or stylus touch.
- Advantages:
  - Direct relationship between hand and pointer location in terms of direction, distance, and speed.
  - Movement is direct, in the same plane as screen.
  - Requires no additional desk space.
  - Stands up well in high-use environments.
- Disadvantages:
  - Finger may obscure part of screen.
  - Finger may be too large for accuracy with small objects.
  - Requires moving the hand far from the keyboard to use.
  - Very fatiguing to use for extended period of time.
  - May soil or damage the screen.
- Design Guidelines:
  - Screen objects should be atleast 3/4” X ¾” in size
  - Object separation should be at least 1/8″.
  - Provide visual feedback in response to activation. Auditory feedback may also be appropriate.
  - When the consequences are destructive, require confirmation after selection to eliminate inadvertent selection.
  - Provide an instructional invitation to begin using.

LIGHT PEN

- Description:
  - A special surface on a screen sensitive to the touch of a special stylus or pen.
- Advantages:
  - Direct relationship between hand and pointer movement in terms of direction, distance, and speed.
  - Movement is direct, in the same plane as screen.
  - Requires minimal additional desk space.

  - Stands up well in high-use environments.
  - More accurate than finger touching.
- Disadvantages:
  - Hand may obscure part of screen.
  - Requires picking it up to use.
  - Requires moving the hand far from the keyboard to use.
  - Very fatiguing to use for extended period of time.

VOICE

- Description:
  - Automatic speech recognition by the computer.
- Advantages:
  - Simple and direct.
  - Useful for people who cannot use a keyboard.
  - Useful when the user's hands are occupied.
- Disadvantages:
  - High error rates due to difficulties in:
    - Recognizing boundaries between spoken words.
    - Blurred word boundaries due to normal speech patterns.
  - Slower throughput than with typing.
  - Difficult to use in noisy environments.
  - Impractical to use in quiet environments.

MOUSE

- Description:
  - A rectangular or dome-shaped, movable, desktop control containing from one to three buttons used to manipulate objects and information on the screen.
  - Movement of screen pointer mimics the mouse movement.
- Advantages:
  - Direct relationship between hand and pointer movement in terms of direction, distance, and speed.
  - Permits a comfortable hand resting position
  - Selection mechanisms are included on mouse.
  - Does not obscure vision of the screen.
- Disadvantages:
  - Movement is indirect, in a plane different from screen.
  - Requires hand to be removed from keyboard.
  - Requires additional desk space.
  - May require long movement distances.
  - Requires a degree of eye-hand coordination.
- Mouse Usage Guidelines

- Provide a —hot zone around small or thin objects that mouse positioning.
- Never use double-clicks or double-drags as the only means operations.
- Do not use mouse plus keystroke combinations.
- Do not require a person to point at a moving target.

A mouse configuration includes one, two or three buttons. The e. Functions. The functions performed by a mouse are Select, Menu, and Adjust.

- The Select function is used to manipulate controls, to select alternatives and data, and to select objects that will be affected by actions that follow. Select is a mouse's most important function and is the function assigned to a one-button mouse. For a multi-button mouse, it is usually assigned to the leftmost button (assuming a right handed operation).
- The Menu function is typically used to request and display a pop-up menu on a screen. A menu appears when the button is depressed within a particular defined area of the screen. This area may be, for example, the entire screen, within a window, or on a window border. This button eliminates the need for a control icon, which must be pointed at and selected. The user, however, must remember that a menu is available.
- The Adjust function extends or reduces the number of items selected. It is the least used of the three functions and is usually assigned last and given the least prominent location on a mouse

Several operations can be performed with a mouse. The first, point, is the movement and positioning of the mouse pointer over the desired screen object. It prepares for a selection or control operation. To press is to hold the button down without releasing it. It identifies the object to be selected.

To click is to press and immediately release a button without moving the mouse. This operation typically selects an item or insertion point, operates a control, or activates an inactive window or control. To double-click is to perform two clicks within a predefined time limit without moving the mouse. It is used as a shortcut for common operations such as activating an icon or opening a file.

To drag is to press and hold the button down, and then move the pointer in the appropriate direction. It identifies a range of objects or moves or resizes items. To double-drag is to perform two clicks and hold the button down, and then move the pointer in the appropriate direction. It identifies a selection by a larger unit, such as a group of words

KEYBOARD

- Description:
  - Standard typewriter keyboard and cursor movement keys.
- Advantages:
  - Familiar.

- Accurate.
- Does not take up additional desk space.
- Very useful for:
  - Entering text and alphanumeric data.
  - Inserting in text and alphanumeric data.
  - Keyed shortcuts—accelerators.
  - Keyboard mnemonics—equivalents.
- Advantageous for:
  - Performing actions when less than three mouse buttons exist.
  - Use with very large screens.
  - Touch typists.

- Disadvantages:
  - Slow for non-touch-typists.
  - Slower than other devices in pointing.
  - Requires discrete actions to operate.
  - No direct relationship between finger or hand movement on the keys and cursor movement on screen in terms of speed and distance.
- Keyboard Guidelines
  - Provide keyboard accelerators.
    - Assign single keys for frequently performed, small-scale tasks.
    - Use standard platform accelerators.
    - Assign Shift-key combinations for actions that extend or are complementary to the actions of the key or key combination used without the Shift-key.
    - Assign Ctrl-key combinations for:
      - ✓ Infrequent actions.
      - ✓ Tasks that represent larger-scale versions of the task assigned to the unmodified key.
  - Provide keyboard equivalents.
    - Use standard platform equivalents.
    - Use the first letter of the item description.
    - If first letter conflicts exist, use:
      - ✓ Another distinctive consonant in the item description.
      - ✓ A vowel in the item description.
  - Provide window navigation through use of keyboard keys.

SELECTING THE PROPER DEVICE-BASED CONTROLS

Guidelines for Selecting the Proper Device-Based Control

- Consider the characteristics of the task.

  - Provide keyboards for tasks involving:
    - Heavy text entry and manipulation.
    - Movement through structured arrays consisting of a few discrete objects.
  - Provide an alternative pointing device for graphical or drawing tasks.
  - The following are some suggested best uses:
    - Mouse—pointing, selecting, drawing, and dragging.
    - Joystick—selecting and tracking.
    - Trackball—pointing, selecting and tracking.
    - Touch screen—pointing and selecting.
    - Graphic tablet—pointing, selecting, drawing, and dragging.
  - Provide touch screens under the following conditions:
    - The opportunity for training is minimal.
    - Targets are large, discrete, and spread out.
    - Frequency of use is low.
    - Desk space is at a premium.
    - Little or no text input requirement exists.
- Consider user characteristics and preferences.
  - Provide keyboards for touch typists.
- Consider the characteristics of the environment.
- Consider the characteristics of the hardware.
- Consider the characteristics of the device in relation to the application.
- Provide flexibility.
- Minimize eye and hand movements between devices.

Keyboard versus Mouse

Speed is obviously one reason. An experienced typist, through kinesthetic memory, has memorized the location of keyboard keys. The keying process becomes exceptionally fast and well learned. The mouse is slower,

Control Research

A survey of the research literature comparing and evaluating different devices yields the following summarization concerning tasks involving pointing and dragging:

- The fastest tools for pointing at stationary targets on screens are the devices that permit direct pointing: the touch screen and light pen. This is most likely due to their high level of eye-hand coordination and because they use an action familiar to people.
- In terms of positioning speed and accuracy for stationary targets, the indirect pointing devices—the mouse, trackball, and graphic tablet, do not differ greatly from one another. The joystick is the slowest, although it is as accurate as the others. Of most importance in selecting one of these devices will be its fit to the user's task and working environment.

- A separate confirmation action that must follow pointer positioning increases pointing accuracy but reduces speed. The mouse offers a very effective design configuration for tasks requiring this confirmation.
- For tracking small, slowly moving targets, the mouse, trackball, and graphic tablet are preferred to the touch screen and light pen because the latter may obscure the user's view of the target.

Another common manipulation task is dragging an object across the screen. Using a mouse, graphic tablet, and trackball for this task, as well as pointing, was studied by MacKenzie, Sellen, and Buxton (1991). They report the following:

- The graphic tablet yielded best performance during pointing.
- The mouse yielded best performance during dragging.
- The trackball was a poor performer for both pointing and dragging, and it had a very high error rate in dragging.

Pointer Guidelines

- The pointer:
  - Should be visible at all times.
  - Should contrast well with its background.
  - Should maintain its size across all screen locations and during movement.
  - The hotspot should be easy to locate and see.
  - Location should not warp (change position).
- The user should always position the pointer.
- The shape of a pointer:
  - Should clearly indicate its purpose and meaning.
  - Should be constructed of already defined shapes.
  - Should not be used for any other purpose other than its already defined meaning.
  - Do not create new shapes for already defined standard functions.
- Use only as many shapes as necessary to inform the user about current location and status. Too many shapes can confuse a person.
- Be conservative in making changes as the pointer moves across the screen.
  - Provide a short "time-out" before making noncritical changes on the screen.
- Animation should not:
  - Distract.
  - Restrict one's ability to interact.

MODULE 5

CHAPTER 9. SELECT THE PROPER KIND OF WINDOWS

Syllabus: Screen based controls – Operable control, Text control, Selection control, Custom control, Presentation control, Windows Tests-prototypes, Kinds of tests

Choose the Proper Screen-Based Controls

- Screen-based controls, often simply called controls and sometimes called widgets, are the elements of a screen that constitute its body.
- By definition, they are graphic objects that represent the properties or operations of other objects. A control may:
  - ✓ Permit the entry or selection of a particular value.
  - ✓ Permit the changing or editing of a particular value.
  - ✓ Display only a particular piece of text, value, or graphic.
  - ✓ Cause a command to be performed.
  - ✓ Possess a contextual pop-up window.
- Three extremely important principles regarding controls should be noted:
  - ✓ A control must: Look the way it works. Work the way it looks.
  - ✓ A control must be used exactly as its design intended.
  - ✓ A control must be presented in a standard manner.
- The look of a control should make it obvious that it is a control. Its design characteristics should signal ―enterability or ―clickability. Microsoft Windows, for example, presents the following simple rules:
  - ✓ Raised elements can be pressed.
  - ✓ Recessed elements cannot be pressed.
  - ✓ Elements on a flat white background can be opened, edited, or moved.

OPERABLE CONTROLS

Operable controls are those that permit the entry, selection, changing, or editing of a particular value, or cause a command to be performed. Classes include buttons, text entry/read-only, selection, combination entry/selection, and other specialized controls.

Buttons

- Description:
  — A square or rectangular-shaped control with a label inside that indicates action to be accomplished.
  — The label may consist of text, graphics, or both.
- Purpose:
  — To start actions.
  — To change properties.

— To display a pop-up menu.

- Advantages:
  — Always visible, reminding one of the choices available.
  — Convenient.
  — Can be logically organized in the work area.
  — Can provide meaningful descriptions of the actions that will be performed.
  — Larger size generally provides faster selection target.
  — Can possess 3-D appearance:
    • Adds an aesthetically pleasing style to the screen.
    • Provides visual feedback through button movement when activated.
  — May permit use of keyboard equivalents and accelerators.
  — Faster than using a two-step menu bar/pull-down sequence.
- Disadvantages:
  — Consumes screen space. — Size limits the number that may be displayed. — Requires looking away from main working area to activate. — Requires moving the pointer to select.
- Proper usage: — Use for frequently used actions that are specific to a window. • To cause something to happen immediately. • To display another window.• To display a menu of options. • To set a mode or property value. • A button comes in three styles. Symbol button

Command Buttons

Figure 7.1 Command buttons.

Figure 7.2 Toolbar buttons without labels.

Figure 7.3 A symbol button.

Command button guidelines include the following.

Usage

- For windows with a menu bar:
  — Use to provide fast access to frequently used or critical commands.
- For windows without a menu bar:
  — Use to provide access to all necessary commands.

Structure

- Provide a rectangular shape with the label inscribed within it.
- Give the button a raised appearance.

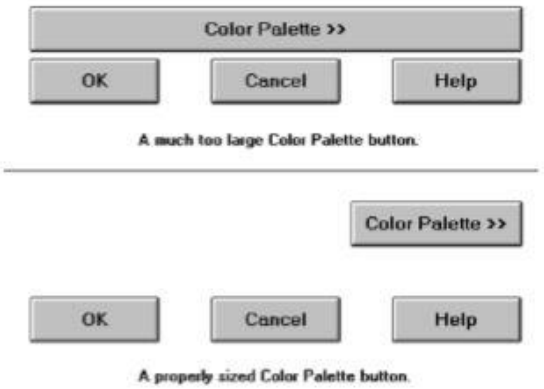
- Maintain consistency in style throughout an application.

Labels

- Use standard button labels when available.
- Provide meaningful descriptions of the actions that will be performed.
- Use single-word labels whenever possible.
  — Use two-three words for clarity, if necessary.
- Use mixed-case letters with the first letter of each significant label word capitalized.
- Display labels:
  — In the regular system font.
  — In the same size font.
- Do not number labels.
- Center the label within the button borders, leaving at least two pixels between the text and the button border.
- Provide consistency in button labeling across all screens.

Size

- Provide as large a button as feasible.
- Maintain consistent button heights and widths.
- Exception: Buttons containing excessively long labels may be wider.



Number

- Restrict the number of buttons on a window to six or fewer.

Location and Layout

- Maintain consistency in button location between windows.
- Never simply “fit” buttons in available space.
- If buttons are for exiting the dialog:
  — Position them centered and aligned horizontally at the bottom.
- If buttons are used for invoking a dialog feature or expanding the dialog:
  — Position them centered and aligned vertically on the right side.
- If a button has a contingent relationship to another control:

— Position it adjacent to the related control.

- If a button has a contingent relationship to a group of controls:
  — Position it at the bottom or to right of related controls.
- If, due to space constraints, exiting and expanding/invoking feature buttons must be placed together:
  — If at the bottom, place exiting buttons to the right, separating the groupings by one button's width.
  — If along the right side, place exiting buttons at the bottom, separating the groupings by one button's height.
- For exiting and expanding/invoking feature buttons, do not:
  — Align with the other screen controls.
  — Present displayed within a line border.
- Provide equal and adequate spacing between adjacent buttons.
- Provide adequate spacing between buttons and the screen body controls.

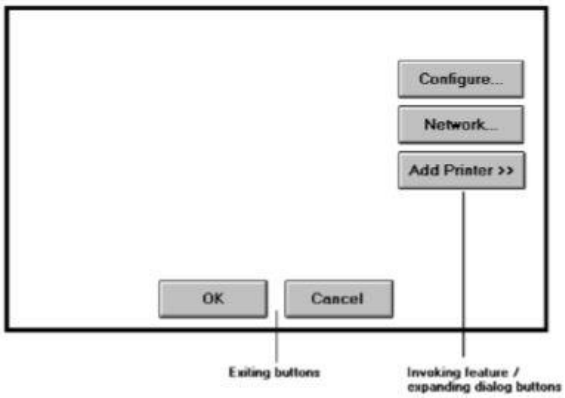


Figure 7.5 Exiting and invoking feature/expanding dialog buttons.

Organization

- Organize standard buttons in the manner recommended by the platform being used.
- For other buttons, organize them in common and customary grouping schemes.
  — For buttons ordered left to right, place those for most frequent actions to the left.
  — For buttons ordered top to bottom, place those for most frequent actions at the top.
- Keep related buttons grouped together.
- Separate potentially destructive buttons from frequently chosen selections.
- Buttons found on more than one window should be consistently positioned.
- The order should never change.
- For mutually exclusive actions, use two buttons; do not dynamically change the text.
- Windows recommends the following:
  - An affirmative action to the left (or above).
  - The default first.
  - OK and Cancel next to each other.
  - Help last, if supported.

Intent Indicators

- When a button causes an action to be immediately performed, no intent indicator is necessary. Ex: **Apply**
- When a button leads to a cascading dialog, include an ellipsis (...) after the label. Ex: **Open..**
- When a button leads to a menu, include a triangle pointing in the direction the menu will appear after the label. Ex: **Menu >**
- When a button leads to an expanding dialog, include a double arrow (>>) with the label. Ex: **Options >>**
- When a button has a contingent relationship to another control that must be indicated, include a single arrow (->) pointing at the control. Ex: ← Clear

Expansion Buttons

- Gray them out after expansion.
- Provide a contraction button, if necessary.
  — Locate it beneath, or to right of, the expansion button.
  — Gray it out when not applicable.

Defaults

- Intent:
  — When a window is first displayed, provide a default action, if practical.
- Selection:
  — A default should be the most likely action:
    - A confirmation.
    - An application of the activity being performed.
    - A positive action such as OK, unless the result is catastrophic.

  — If a destructive action is performed (such as a deletion), the default should be Cancel.
- Presentation: — Indicate the default action by displaying the button with a bold or double border.
- Procedures:
  — The default can be changed as the user interacts with the window.
  — When the user navigates to a button, it can temporarily become the default.
  — Use the Enter key to activate a default button.
  — If another control requires use of the Enter key, temporarily disable the default while the focus is on the other control.
  — Permit double-clicking on a single selection control in a window to also carry out the default command.

Unavailable Choices

- Temporarily unavailable choices should be dimmed or grayed out.

Keyboard Equivalents and Accelerators

- Equivalents:
  — Assign a keyboard equivalent mnemonic to each button to facilitate keyboard selection.
  — The mnemonic should be the first character of the button's label.
    - If duplication exists in first characters, for duplicate items, use another character in the label.
    - Preferably, choose the first succeeding consonant.
  — Designate the mnemonic character by underlining it. Ex: **Apply**
  — Maintain the same mnemonic on all identical buttons on other screens.
- Accelerators:
  — Assign a keyboard accelerator to each button to facilitate keyboard selection.

Scrolling

- If a window can be scrolled, do not scroll the command buttons.
  — Exception: if the screen cannot scroll independently of the buttons.
- Use buttons to move between multipage forms, not scroll bars.
  — Label buttons Next and Previous.

Button Activation

- Pointing:
  — Highlight the button in some visually distinctive manner when the pointer is resting on it and the button is available for selection.
- Activation:
  — Call attention to the button in another visually distinctive manner when it has been activated or pressed.
  — If a button can be pressed continuously, permit the user to hold the mouse button down and repeat the action.

Toolbars

Toolbars are compilations of commands, actions, or functions, usually graphical in structure but sometimes textual, grouped together for speedy access. Toolbars may also be called button bars, control bars, or access bars. Specialized toolbars may also be referred to as ribbons, toolboxes, or palettes. Toolbars may also appear in palette windows.

Usage

- To provide easy and fast access to most frequently used commands or options across multiple screens.
- To invoke a sub application within an application.
- To use in place of certain menu items.

Structure

- Images:
  — Provide buttons of equal size.
  — Create a meaningful and unique icon.
    - Design them using icon design guidelines.
  — Center the image within the button.
  — Give the button a raised appearance.
  — Ensure that toolbar images are discernible from Web page graphical images.
- Text:
  — Create a meaningful label, adhering to label guidelines for command buttons.
  — Create toolbar buttons of equal size, following the size guidelines recently described.
- Consistency:
  — Use the same icon throughout an application and between applications.

Size

- Button:
  — 24 (w) by 22 (h) pixels, including border.
  — 32 (w) by 30 (h) pixels, including border.
  — Larger buttons can be used on high-resolution displays.
- Label:
  — 16 (w) by 16 (h) pixels.
  — 14 (w) by 24 (h) pixels.
- Default:
  — Provide the smaller size as the default size with a user option to change it.
- Image:
  — Center the image in the button.

Organization

- Order the buttons based on common and customary grouping schemes.
  — For buttons ordered left to right, place those for the most frequently used actions to the left.
  — For buttons ordered top to bottom, place those for the most frequently used actions at the top.
- Keep related buttons grouped together.
- Separate potentially destructive buttons from frequently chosen selections.
- Permit user reconfiguration of button organization.

Location

- Position main features and functions bar horizontally across top of window just below menu bar.

- Position subtask and sub features bars along sides of window.
- Permit the location of the bar to be changed by the user.
- Permit display of the bar to be turned on or off by the user.
  — Also provide access through standard menus.

Active Items

- Make only currently available toolbar items available.
- Temporarily not available items may be displayed grayed out.

Customization

- Permit toolbars to be turned off by the user.
- Allow the customizing of toolbars.
  — Provide a default, however.

Keyboard Equivalents and Accelerators

- Equivalents:
  — Assign keyboard equivalents to facilitate keyboard selection.
  — Maintain the same mnemonic on all identical buttons on all screens.
- Accelerators:
- — Assign a keyboard accelerator to facilitate keyboard selection.

Button Activation

- Pointing:
  — Highlight the button in some visually distinctive manner when the pointer is resting on it and the button is available for selection.
- Activation:
  — Call attention to the button in another visually distinctive manner when it has been activated or pressed.

TEXT ENTRY/READ-ONLY CONTROLS

A Text Entry/Read-Only control contains text that is exclusively entered or modified through the keyboard. It may also contain entered text being presented for reading or display purposes only.

Text Boxes

- Description:
  — A control, usually rectangular in shape, in which:
    - Text may be entered or edited.
    - Text may be displayed for read-only purposes.
  — Usually possesses a caption describing the kind of information contained within it.

— An outline field border:
  - Is included for enterable/editable text boxes.
  - Is not included for read-only text boxes.

— Two types exist:
  - Single line.
  - Multiple line.

— When first displayed, the box may be blank or contain an initial value.

- Purpose:
  — To permit the display, entering, or editing of textual information.
  — To display read-only information.
- Advantages:
  — Very flexible.
  — Familiar.
  — Consumes little screen space.
- Disadvantages:
  — Requires use of typewriter keyboard.
  — Requires user to remember what must be keyed.
- Proper usage:
  — Most useful for data that is:
    - Unlimited in scope.
    - Difficult to categorize.
    - Of a variety of different lengths.
  — When using a selection list is not possible.

Types of text box

- Two types of text boxes exist. One consists of a rectangular box into which information is typed. It may also be referred to as an edit control.
- The second is also rectangular in shape but contains text displayed purely for read-only purposes. The former type has historically been referred to as an entry field, the latter as an inquiry or display field.

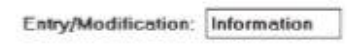


Figure 7.14 Text boxes.

Two forms of Text Box

Single-Line and Multiple-Line Text Boxes

- Single line:
  — Description:
    - A control consisting of no more than one line of text.
  — Purpose:
    - To make textual entries when the information can be contained within one line of the screen.
  — Typical uses:
    - Typing the name of a file to save.
    - Typing the path of a file to copy.
    - Typing variable data on a form.
    - Typing a command.
- Multiple line:
  — Description:
    - A control consisting of a multiline rectangular box for multiple lines of text.
  — Purpose:
    - To type, edit, and read passages of text.
  — Typical uses:
    - Creating or reading an electronic mail message.
    - Displaying and editing text files.

Captions

- Structure and size:
  — Provide a descriptive caption to identify the kind of information to be typed, or contained within, the text box.
  — Use a mixed-case font.
  — Display the caption in normal intensity or in a colour of moderate brightness.
- Formatting:
  — Single fields:
    - Position the field caption to the left of the text box.
      — Place a colon (:) immediately following the caption.
      — Separate the colon from the text box by one space. Ex: Composition:
    - Alternately, the caption may be placed above the text box.
      — Place a colon (:) immediately following the caption.
      — Position above the upper-left corner of the box, flush with the left edge.
      — Multiple occurrence fields:
      Composition:
    - For entry/modification text boxes:
      — Position the caption left-justified one line above the column of entry fields.
      Offices:

- For display/read-only boxes:

  — If the data field is long and fixed-length, or the displayed data is about the same length, center the caption above the displayed text box data.

  Ex: Date:

07/17/2018
07/18/2019
07/19/2020

  — If the data displayed is alphanumeric, short, or quite variable in length, left-justify the caption above the displayed text box data.

  Ex: Location:

Preran
Sushmitha
Sanskar
Riya

  — If the data field is numeric and variable in length, right-justify the caption above the displayed text box data.

  Ex: Balances:

12,643,24.0897
56.84
35,23,545.98
434.345

Fields

- Structure:

  — Identify entry/modification text boxes with a line border or reverse polarity rectangular box.
  - To visually indicate that it is an enterable field, present the box in a recessed manner.
  - Present display/read-only text boxes on the window background.

  — Break up long text boxes through incorporation of slashes (/), dashes (-), spaces, or other common delimiters.

  Ex: Date :

  Date: / /
- Size:

  — Size to indicate the approximate length of the field.

— Text boxes for fixed-length data must be large enough to contain the entire entry.
— Text boxes for variable-length data must be large enough to contain the majority of the entries.
— Where entries may be larger than the entry field, scrolling must be provided to permit keying into, or viewing, the entire field.
— Employ word wrapping for continuous text in multiple-line text boxes.

- Highlighting:
  — Call attention to text box data through a highlighting technique.
  - Higher intensity.
  - If colour is used, choose one that both complements the screen background and contrasts well with it.
- Unavailable fields:
  — Gray-out temporarily unavailable text boxes.
- Fonts:
  — To support multiple fonts, use a Rich-Text Box.

SELECTION CONTROLS

- A selection control presents on the screen all the possible alternatives, conditions, or choices that may exist for an entity, property, or value.
- The relevant item or items are selected from those displayed.
- Selection controls include radio buttons, check boxes, list boxes, drop-down/pop-up list boxes, and palettes.

Radio Buttons

- Description:
  — A two-part control consisting of the following: • Small circles, diamonds, or rectangles. • Choice descriptions.
  — When a choice is selected: • The option is highlighted. • Any existing choice is automatically unhighlighted and deselected.
- Purpose:
  — To set one item from a small set of mutually exclusive options (2 to 8).
- Advantages:
  — Easy-to-access choices.
  — Easy-to-compare choices.
  — Preferred by users.
- Disadvantages:
  — Consume screen space.
  — Limited number of choices.
- Proper usage:
  — For setting attributes, properties, or values.

— For mutually exclusive choices (that is, only one can be selected).
— Where adequate screen space is available.
— Most useful for data and choices that are:
  - Discrete.
  - Small and fixed in number.
  - Not easily remembered.
  - In need of a textual description to meaningfully describe the alternatives.
  - Most easily understood when the alternatives can be seen together and compared to one another.
  - Never changed in content.
— Do not use:
  - For commands.
  - Singly to indicate the presence or absence of a state.

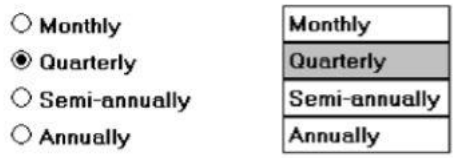


Figure 7.24 Radio buttons.

Figure 7.25 Radio buttons.

Choice Descriptions

- Provide meaningful, fully spelled-out choice descriptions clearly describing the values or effects set by the radio buttons.
- Display in a single line of text.
- Display using mixed-case letters, using the sentence style.
- Position descriptions to the right of the button. Separate them by at least one space from the button.
- When a choice is conditionally unavailable for selection, display the choice description grayed out or dimmed.
- Include a none choice if it adds clarity.

Size

- Show a minimum of two choices, a maximum of eight.

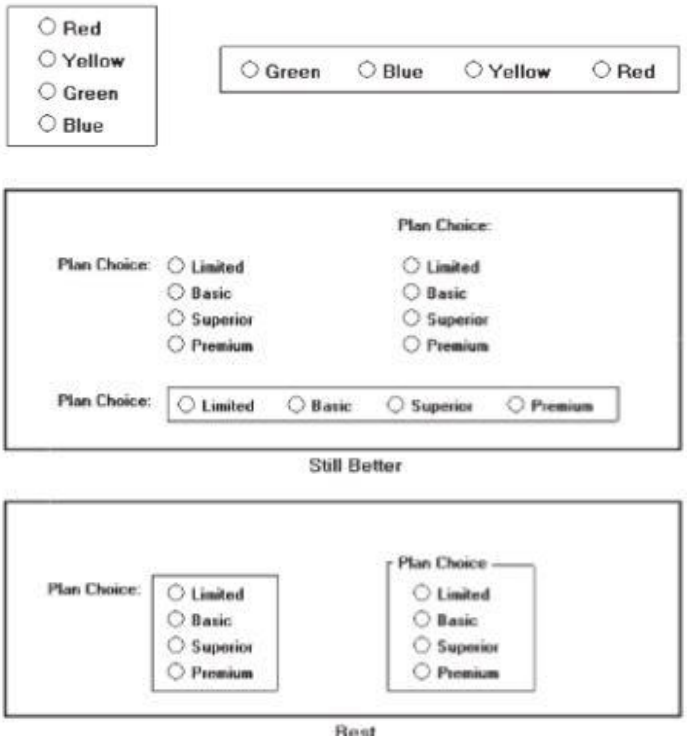
Defaults

- When the control possesses a state or affect that has been predetermined to have a higher probability of selection than the others, designate it as the default and display its button filled in.
- When the control includes choices whose states cannot be predetermined, display all the buttons without setting a dot, or in the indeterminate state.

- When a multiple selection includes choices whose states vary, display the buttons in another unique manner, or in the mixed value state.

Structure

- A columnar orientation is the preferred manner of presentation.
- Left-align the buttons and choice descriptions.
- If vertical space on the screen is limited, orient the buttons horizontally.
- Provide adequate separation between choices so that the buttons are associated with the proper description.
  — A distance equal to three spaces is usually sufficient.
- Enclose the buttons in a border to visually strengthen the relationship they possess.



Organization

- Arrange selections in expected order or follow other patterns such as frequency of occurrence, sequence of use, or importance.
  — For selections arrayed top to bottom, begin ordering at the top.
  — For selections arrayed left to right, begin ordering at the left.
- If, under certain conditions, a choice is not available, display it subdued or less brightly than the available choices.

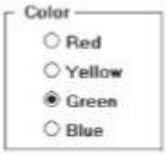
Related Control

- Position any control related to a radio button immediately to the right of the choice description.

- If the radio button choice description also acts as the label for the control that follows it, end the label with an arrow (>).

Captions

- Structure:
  — Provide a caption for each radio button control.
    - Exception: In screens containing only one radio button control, the screen title may serve as the caption.
- Display:
  — Fully spelled out.
  — In mixed-case letters, capitalizing the first letter of all significant words.
- Columnar orientation:
  — With a control border, position the caption:
    - Upper-left-justified within the border.



    - Alternately, the caption may be located to the left of the topmost choice description.

  — Without an enclosing control border, position the caption:
    - Left-justified above the choice descriptions, separated by one space line.

    - Alternately, the caption may be located to the left of the topmost choice description.

- Horizontal orientation:
  — Position the caption to the left of the choice descriptions.

    - Alternately, with an enclosing control border, left-justified within the border.

— Be consistent in caption style and orientation within a screen.

Keyboard Equivalents

- Assign a keyboard mnemonic to each choice description.
- Designate the mnemonic by underlining the applicable letter in the choice description.

Selection Method and Indication

- Pointing:
  — The selection target area should be as large as possible.
    - Include the button and the choice description text.

  — Highlight the selection choice in some visually distinctive way when the cursor's resting on it and the choice is available for selection.
    - This cursor should be as long as the longest choice description plus one space at each end. Do not place the cursor over the small button.

- Activation:
  — When a choice is selected, distinguish it visually from the unselected choices. • A radio button should be filled in with a solid dark dot or made to look depressed or higher through use of a shadow.
  — When a choice is selected, any other selected choice must be deselected.
- Defaults:
  — If a radio button control is displayed that contains a choice previously selected or a default choice, display the selected choice as set in the control

Check Boxes

- Description:
  — A two-part control consisting of a square box and choice description.
  — Each option acts as a switch and can be either "on" or "off"
    - When an option is selected (on), a mark such as an "X" or "check" appears within the square box, or the box is highlighted in some other manner.
    - Otherwise the square box is unselected or empty (off).

  — Each box can be:
    - Switched on or off independently.
    - Used alone or grouped in sets.
- Purpose:
  — To set one or more options as either on or off.

- Advantages
  — Easy-to-access choices.
  — Easy-to-compare choices.
  — Preferred by users.
- Disadvantages:
  — Consume screen space.
  — Limited number of choices.
  — Single check boxes difficult to align with other screen controls.
- Proper usage:
  — For setting attributes, properties, or values.
  — For nonexclusive choices (that is, more than one can be selected).
  — Where adequate screen space is available.
  — Most useful for data and choices that are:
    - Discrete.
    - Small and fixed in number.
    - Not easily remembered.
    - In need of a textual description to describe meaningfully.
    - Most easily understood when the alternatives can be seen together and compared to one another.
    - Never changed in content.
  — Can be used to affect other controls.
  — Use only when both states of a choice are clearly opposite and unambiguous.

Figure 7.39 Check boxes.

Figure 7.40 Check boxes.

Choice Descriptions

- Provide meaningful, fully spelled-out choice descriptions clearly describing the values or effects set by the check boxes.
- Display them in a single line of text.
- Display them using mixed-case letters in sentence style.

- Position descriptions to the right of the check box. Separate by at least one space from the box.
- When a choice is unavailable for selection under a certain condition, display the choice description visually dimmed.

Size

- Show a minimum of one choice, a maximum of eight.

Defaults

- When the control possesses a state or affect that has been preset, designate it as the default and display its check box marked.
- When a multiple selection includes choices whose states vary, display the buttons in another unique manner, or the mixed value state.

Structure

- Provide groupings of related check boxes.
- A columnar orientation is the preferred manner of presentation for multiple related check boxes.
- Left-align the check boxes and choice descriptions.
- If vertical space on the screen is limited, orient the boxes horizontally.
- Provide adequate separation between boxes so that the buttons are associated with the proper description.
  — A distance equal to three spaces is usually sufficient.
- Enclose the boxes in a border to visually strengthen the relationship they possess.

Organization

- Arrange selections in logical order or follow other patterns such as frequency of occurrence, sequence of use, or importance.
  — For selections arrayed top to bottom, begin ordering at the top.
  — For selections arrayed left to right, begin ordering at the left.
- If, under certain conditions, a choice is not available, display it subdued or less brightly than the available choices.

Related Control

- Position any control related to a check box immediately to the right of the choice description.
  — If a the check box choice description also acts as the label for the control that follows it , end the label with an arrow (>).

Captions and Keyboard Equivalents

Same as Radio Button

Selection Method and Indication

- Pointing:
  — The selection target area should be as large as possible.
    - Include the check box and the choice description text.

  — Highlight the selection choice in some visually distinctive way when the cursor's resting on it and the choice is available for selection.
    - This cursor should be as long as the longest choice description plus one space at each end. Do not place the cursor over the check box.
- Activation:
  — When a choice is selected, distinguish it visually from the non-selected choices.
    - A check box should be filled in or made to look depressed or higher through use of a shadow.
- Defaults:
  — If a check box is displayed that contains a choice previously selected or default choice, display the selected choice as set in the control.
- Select/deselect all:
  — Do not use Select All and Deselect All check boxes.
- Mixed-value state:
  — When a check box represents a value, and a multiple selection encompasses multiple value occurrences set in both the on and off state, display the check box in a mixed value state.

    - Fill the check box with another easily differentiable symbol or pattern.

  — Toggle the check box as follows:
    - Selection 1: Set the associated value for all elements. Fill the check box with an "X" or "check"
    - Selection 2: Unset the value for all associated elements. Blank-out the check box.
    - Selection 3: Return all elements to their original state. Fill the check box with the mixed value symbol or pattern.

Palettes

- Description:
  — A control consisting of a series of graphical alternatives. The choices themselves are descriptive, being composed of colours, patterns, or images.
  — In addition to being a standard screen control, a palette may also be presented on a pull-down or pop-up menu or a toolbar.

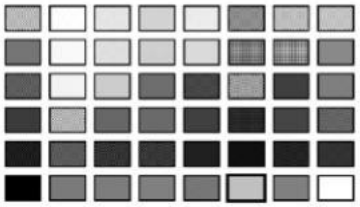


Figure 7.55 Palette.

- Purpose:
  — To set one of a series of mutually exclusive options presented graphically or pictorially.
- Advantages:
  — Pictures aid comprehension.
  — Easy-to-compare choices.
  — Usually consume less screen space than textual equivalents.
- Disadvantages:
  — A limited number of choices can be displayed.
  — Difficult to organize for scanning efficiency.
  — Requires skill and time to design meaningful and attractive graphical representations.
- Proper usage:
  — For setting attributes, properties, or values.
  — For mutually exclusive choices (that is, only one can be selected).
  — Where adequate screen space is available.
  — Most useful for data and choices that are:
    - Discrete.
    - Frequently selected.
    - Limited in number.
    - Variable in number.
    - Not easily remembered.
    - Most easily understood when the alternatives may be seen together and compared to one another.
    - Most meaningfully represented pictorially or by example.
    - Can be clearly represented pictorially.
    - Rarely changed in content.
  — Do not use:
    - Where the alternatives cannot be meaningfully and clearly represented pictorially.
    - Where words are clearer than images.
    - Where the choices are going to change.

- Provide meaningful, accurate, and clear illustrations or representations of choices.
- Create images large enough to: — Clearly illustrate the available alternatives. — Permit ease in pointing and selecting.
- Create images of equal size.
- Always test illustrations before implementing them.

Size

- Present all available alternatives within the limits imposed by:
  — The size of the graphical representations.
  — The screen display's capabilities.

Layout

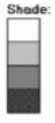
- Create boxes large enough to:
  — Effectively illustrate the available alternatives.
  — Permit ease in pointing and selecting.
- Create boxes of equal size.
- Position the boxes adjacent to, or butted up against, one another.
- A columnar orientation is the preferred manner.
- If vertical space on the screen is limited, orient the choices horizontally.

Organization

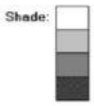
- Arrange palettes in expected or normal order.
  — For palettes arrayed top to bottom, begin ordering at the top.
  — For palettes arrayed left to right, begin ordering at the left.
- If an expected or normal order does not exist, arrange choices by frequency of occurrence, sequence of use, importance, or alphabetically (if textual).
- If, under certain conditions, a choice is not available, display it subdued or less brightly than the other choices.

Captions

- Provide a caption for each palette.
  — On screens containing only one palette, the screen title may serve as the caption.
- Display the caption fully spelled out using mixed-case letters.
- Columnar orientation:
  — The field caption may be positioned left-aligned above the palette.



  — Alternately, the caption may be positioned to the left of the topmost alternative.



- Horizontal orientation:
  — The field caption may be positioned above the palette.

  — Alternately, the caption may be positioned to the left of the alternatives.

Selection Method and Indication

- Pointing:
  — Highlight the choice in some visually distinctive way when the pointer or cursor is resting on it and the choice is available for selection.
- Activation:
  — When a choice is selected, distinguish it visually from the unselected choices by highlighting it in a manner different from when it is pointed at, or by placing a bold border around it.
- Defaults:
  — If a palette is displayed with a choice previously selected or a default choice, display the currently active choice in the manner used when it was selected.

List Boxes

- Description:
  — A permanently displayed box-shaped control containing a list of attributes or objects from which:
    - A single selection is made (mutually exclusive), or
    - Multiple selections are made (non-mutually-exclusive).
  — The choice may be text, pictorial representations, or graphics.
  — Selections are made by using a mouse to point and click.
  — Capable of being scrolled to view large lists of choices.
  — No text entry field exists in which to type text.
  — A list box may be may be associated with a summary list box control, which allows the selected choice to be displayed or an item added to the list.

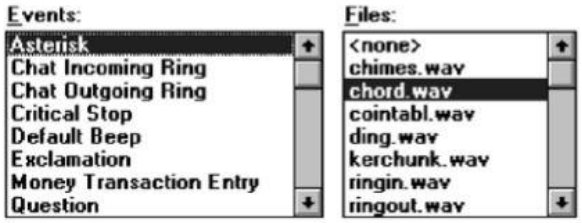


Figure 7.60 List boxes.

- Purpose:
  — To display a collection of items containing:
    - Mutually exclusive options.
    - Non-mutually-exclusive options.
- Advantages:
  — Unlimited number of choices.
  — Reminds users of available options.
  — Box always visible.
- Disadvantages:
  — Consumes screen space.
  — Often requires an action (scrolling) to see all list choices.
  — The list content may change, making it hard to find items.
  — The list may be ordered in an unpredictable way, making it hard to find items.
- Proper usage:
  — For selecting values or setting attributes.
  — For choices that are:
    - Mutually exclusive (only one can be selected).
    - Non-mutually-exclusive (one or more may be selected).
  — Where screen space is available.
  — For data and choices that are:
    - Best represented textually.
    - Not frequently selected.
    - Not well known, easily learned, or remembered.
    - Ordered in an unpredictable fashion.
    - Frequently changed.
    - Large in number.
    - Fixed or variable in list length.

    —When screen space or layout considerations make radio buttons or check boxes impractical.

List Box General Guidelines

Selection Descriptions

- Clearly and meaningfully describe the choices available. Spell them out as fully as possible.
  — Graphical representations must clearly represent the options.
- Present in mixed case, using the sentence style structure.
- Left-align into columns.

List Size

- Not actual limit in size.
- Present all available alternatives.
- Require no more than 40 page-downs to search a list.
  — If more are required, provide a method for using search criteria or scoping the options.

Box Size

- Must be long enough to display 6 to 8 choices without requiring scrolling.
  — Exceptions:
    - If screen space constraints exist, the box may be reduced in size to display at least three items.
    - If it is the major control within a window, the box may be larger.

  — If more items are available than are visible in the box, provide vertical scrolling to display all items.
- Must be wide enough to display the longest possible choice.

  — When box cannot be made wide enough to display the longest entry:
    - Make it wide enough to permit entries to be distinguishable, or,
    - Break the long entries with an ellipsis (...) in the middle, or,
    - Provide horizontal scrolling.

Organization

- Order in a logical and meaningful way to permit easy browsing.
  — Consider using separate controls to enable the user to change the sort order or filter items displayed in the list.
- If a particular choice is not available in the current context, omit it from the list.
  — Exception: If it is important that the existence and unavailability of a particular list item be communicated, display the choice dimmed or grayed out instead of deleting it.

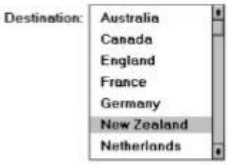
Layout and Separation

- Enclose the choices in a box with a solid border.
  — The border should be the same colour as the choice descriptions.
- Leave one blank character position between the choice descriptions and the left border.
- Leave one blank character position between the longest choice description in the list and the right border, if possible.

Captions

- Use mixed-case letters.
- The preferred position of the control caption is above the upper-left corner of the list box.

- Alternately, the caption may be located to the left of the topmost choice description.



- Be consistent in caption style and orientation within a screen, and related screens.

Disabling

- When a list box is disabled, display its caption and show its entries as grayed out or dimmed.

Selection Method and Indication

- Pointing:
  — Highlight the selection choice in some visually distinctive way when the pointer or cursor is resting on it and the choice is available for selection.
- Selection:
  — Use a reverse video or reverse colour bar to surround the choice description when it is selected.
  — The cursor should be as wide as the box itself.

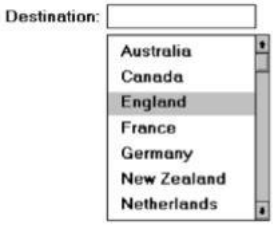
— Mark the selected choice in a distinguishing way.

- Activation:
  — Require the pressing of a command button when an item, or items, is selected.

Single-Selection List Boxes

- Purpose:
  — To permit selection of only one item from a large listing.
- Design guidelines:
  — Related text box
  • If presented with an associated text box control:
  — Position the list box below and as close as possible to the text box.
  — The list box caption should be worded similarly to the text box caption.

— If the related text box and the list box are very close in proximity, the caption may be omitted from the list box.



— Use the same background colour for the text box as is used in the list box.

- Defaults:
  When the list box is first displayed:
  — Present the currently active choice highlighted or marked with a circle or d diamond to the left of the entry.
  — If a choice has not been previously selected, provide a default choice and display it in the same manner that is used in selecting it.
  — If the list represents mixed values for a multiple selection, do not highlight an entry.

- Other:
  —Follow other relevant list box guidelines.

Extended and Multiple-Selection List Boxes

- Purpose:
  — To permit selection of more than one item in a long listing.
    - Extended list box: Optimized for individual item or range selection.
    - Multiple-selection list box: Optimized for independent item selection.
- Design guidelines:
  — Selection indication:
  • Mark the selected choice with an X or check mark to the left of the entry.

  • Consider providing a summary list box.
  — Position it to the right of the list box.
  — Use the same colours for the summary list box as are used in the list box.

  — Provide command buttons to Add (one item) or Add All (items) to the summary list box and Remove (one item) or Remove All (items) from the summary list box.
- Consider providing a display-only text control indicating how many choices have been selected.
  — Position it justified upper-right above the list box.

  — Select all and Deselect All buttons
  — Provide command buttons to accomplish fast Select All and Deselect All actions, when these actions must be frequently or quickly performed.
  — Defaults:
  • When the list box is first displayed:
  — Display the currently active choices highlighted.

— Mark with an X or check mark to the left of the entry.
— If the list represents mixed values for a multiple selection, do not highlight an entry.
— Other:
• Follow other relevant list box guidelines.

List View Controls

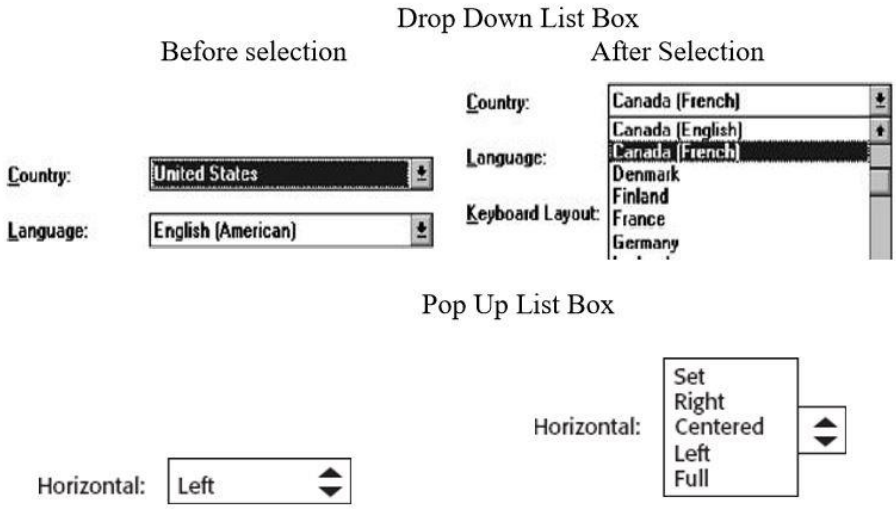
- Description:
  — A special extended-selection list box that displays a collection of items, consisting of an icon and a label.
  — The contents can be displayed in four different views:
    - Large Icon: Items appear as a full-sized icon with a label below.
    - Small Icon: Items appear as a small icon with label to the right.
    - List: Items appear as a small icon with label to the right.
      — Arrayed in a columnar, sorted layout.
      • Report: Items appear as a line in a multicolumn format.
      — Leftmost column includes icon and its label.
      — Subsequent columns include application-specific information.
- Purpose and usage:
  — Where the representation of objects as icons is appropriate.
  — To represent items with multiple columns of information.

Drop-down/Pop-up List Boxes

- Description
  — A single rectangular control that shows one item with a small button to the right side.
  • The button provides a visual cue that an associated selection box is available but hidden.
  — When the button is selected, a larger associated box appears, containing a list of choices from which one may be selected.
  — Selections are made by using the mouse to point and click.
  — Text may not be typed into the control.
- Purpose:
  — To select one item from a large list of mutually exclusive options when screen space is limited.
- Advantages:
  — Unlimited number of choices.
  — Reminds users of available options.
  — Conserves screen space.
- Disadvantages:
  — Requires an extra action to display the list of choices.
  — When displayed, all choices may not always be visible, requiring scrolling.

— The list may be ordered in an unpredictable way, making it hard to find items.

- Proper usage:
  — For selecting values or setting attributes.
  — For choices that are mutually exclusive (only one can be selected).
  — Where screen space is limited.
  — For data and choices that are:
    - Best represented textually.
    - Infrequently selected.
    - Not well known, easily learned, or remembered.
    - Ordered in a unpredictable fashion.
    - Large in number.
    - Variable or fixed in list length.
  — Use drop-down/pop-up lists when:
    - Screen space or layout considerations make radio buttons or singleselection list boxes impractical.
    - The first, or displayed, item will be selected most of the time.
  — Do not use a drop-down list if it important that all options be seen together.



Prompt Button

- Provide a visual cue that a box is hidden by including a downward pointing arrow, or other meaningful image, to the right side of the selection field.
  — Position the button directly against, or within, the selection field.

Selection Descriptions

- Clearly and meaningfully describe the choices available. Spell them out as fully as possible.
  — Graphical representations must clearly represent the options.
  — Left-align them in columns.

— Display the descriptions using mixed-case letters.

List Size

- Not limited in size.
- Present all available alternatives.

Box Size

- Long enough to display 6 to 8 choices without scrolling.
  — If more than eight choices are available, provide vertical scrolling to display all items.
- Wide enough to display the longest possible choice.
- When a box cannot be made wide enough to display the longest entry:
  — Make it wide enough to permit entries to be distinguishable, or,
  — Break long entries with ellipses (...) in the middle, or,
  — Provide horizontal scrolling.

Organization

- Order in a logical and meaningful way to permit easy browsing.
- If a particular choice is not available in the current context, omit it from the list.
  — Exception: If it is important that the existence and unavailability of a particular list item be communicated, display the choice dimmed or grayed out instead of deleting it.

Layout and Separation

- Enclose the choices in a box composed of a solid line border.
  — The border should be the same colour as the choice descriptions.
  — Leave one blank character position between the choices and the left border.
  — Leave one blank character position between the longest choice description in the list and the right border, if possible.

Captions

- Display using mixed-case letters.
- Position the caption to the left of the box.
  — Alternately, it may be positioned left-justified above the box.

Defaults

- When the drop-down/pop-up listing is first presented, display the currently set value.
- If a choice has not been previously selected, provide a default choice.

Disabling

- When a drop-down/pop-up list box is disabled, display its caption and entries as disabled or dimmed.

Selection Method and Indication

- Pointing:
  — Highlight the selection choice in some visually distinctive way when the pointer or cursor is resting on it and the choice is available for selection.
- Activation:
  — Close the drop-down/pop-up list box when an item is selected.

COMBINATION ENTRY/SELECTION CONTROLS

- It is possible for a control to possess the characteristics of both a text field and a selection field.
- The types of combination entry/selection fields are spin boxes, attached combination boxes, and drop-down/pop-up combination boxes.

Spin Boxes

- Description:
  — A single-line field followed by two small, vertically arranged buttons.
    - The top button has an arrow pointing up.
    - The bottom button has an arrow pointing down.
  — Selection/entry is made by:
    - Using the mouse to point at one of the directional buttons and clicking. Items will change by one unit or step with each click.
    - Keying a value directly into the field itself.
- Purpose:
  — To make a selection by either scrolling through a small set of meaningful predefined choices or typing text.
- Advantages:
  — Consumes little screen space.
  — Flexible, permitting selection or typed entry.
- Disadvantages:
  — Difficult to compare choices.
  — Can be awkward to operate.
  — Useful only for certain kinds of data.
- Proper usage:
  — For setting attributes, properties, or values.
  — For mutually exclusive choices (only one can be selected).
  — When the task requires the option of either key entry or selection from a list.
  — When the user prefers the option of either key entry or selection from a list.
  — Where screen space is limited.

— Most useful for data and choices that are:
- Discrete.
- Infrequently selected.
- Well known, easily learned or remembered, and meaningful.
- Ordered in a predictable, customary, or consecutive fashion.
- Infrequently changed.
- Small in number.
- Fixed or variable in list length.

List Size

- Keep the list of items relatively short.
- To reduce the size of potentially long lists, break the listing into subcomponents, if possible.

List Organization

- Order the list in the customary, consecutive, or expected order of the information contained within it.
  — Ensure that the user can always anticipate the next (not-yet-visible) choice.
- When first displayed, present a default choice in the box.

Other Spin Box Guidelines

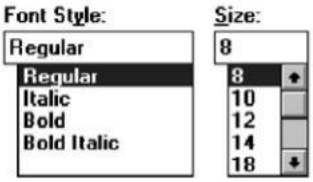
- Box size:
  — The spin box should be wide enough to display the longest entry or choice.
- Caption:
  — Display it using mixed-case letters.
  — Position the caption to the left of the box.
  - Alternately, it may be positioned left-justified above the box.
- Entry and selection methods:
  — Permit completion by:
  - Typing directly into the box.
  - Scrolling and selecting with a mouse.
  - Scrolling and selecting with the up/down arrow keys.
  — For alphabetical values:
  - Move down the order using the down arrow.
  - Move up the order using the up arrow.
  - For numeric values or magnitudes:
  - Show a larger value using the up arrow.

- Show a smaller value using the down arrow.

Combo Boxes

- Description:
  — A single rectangular text box entry field, beneath which is a larger rectangular list box (resembling a drop-down list box) displaying a list of options.
  — The text box permits a choice to be keyed within it.
  — The larger box contains a list of mutually exclusive choices from which one may be selected for placement in the entry field.
    - Selections are made by using a mouse to point and click.
  — As text is typed into the text box, the list scrolls to the nearest match.
  — When an item in the list box is selected, it is placed into the text box, replacing the existing content.
  — Information keyed may not necessarily have to match the list items.
- Purpose:
  — To allow either typed entry in a text box or selection from a list of options in a permanently displayed list box attached to the text box.
- Advantages:
  — Unlimited number of entries and choices.
  — Reminds users of available options.
  — Flexible, permitting selection or typed entry.
  — Entries not necessarily restricted to items selectable from list box.
  — List box always visible.
- Disadvantages:
  — Consumes some screen space.
  — All list box choices not always visible, requiring scrolling.
  — Users may have difficulty recalling sufficient information to type entry, making text box unusable.
  — The list may be ordered in an unpredictable way, making it hard to find items.
- Proper usage:
  — For entering or selecting objects or values or setting attributes.
  — For information that is mutually exclusive (only one can be entered or selected).
  — When users may find it practical to, or prefer to, type information rather than selecting it from a list.
  — When users can recall and type information faster than selecting it from a list.
  — When it is useful to provide the users a reminder of the choices available.
  — Where data must be entered that is not contained in the selection list.
  — Where screen space is available.
  — For data and choices that are:
    - Best represented textually.

- Somewhat familiar or known.
- Ordered in an unpredictable fashion.
- Frequently changed.
- Large in number.
- Variable or fixed in list length.



Combo Box Guidelines

For the text box entry field, see "Text Box/Single Line" guidelines. For the list box, see "Drop-down/Pop-up List Box" guidelines.

Drop-down/Pop-up Combo Boxes

- Description:
  — A single rectangular text box with a small button to the side and an associated hidden list of options.
    - The button provides a visual cue that an associated selection box is available but hidden.

  — When requested, a larger associated rectangular box appears, containing a scrollable list of choices from which one is selected.
  — Selections are made by using the mouse to point and click.
  — Information may also be keyed into the field.
  — As text is typed into the text box, the list scrolls to the nearest match.
  — When an item in the list box is selected, it is placed into the text box, replacing the existing content.
  — The information keyed does not necessarily have to match list items.
  — Combines the capabilities of both a text box and a drop-down/pop-up list box.
- Purpose:
  — To allow either typed entry or selection from a list of options in a list box that may be closed and retrieved as needed.
- Advantages:
  — Unlimited number of entries and choices.
  — Reminds users of available options.
  — Flexible, permitting selection or typed entry.
  — Entries not restricted to items selectable from list box.
  — Conserves screen space.
- Disadvantages:

— Requires an extra step to display the list of choices.
— When displayed, all box choices may not always be visible, requiring scrolling.
— User may have difficulty in recalling what to type.
— The list content may change, making it hard to find items.
— The list may be ordered in an unpredictable way, making it hard to find items.

- Proper usage:
  — For entering or selecting objects or values or setting attributes.
  — For information that is mutually exclusive (only one can be entered or selected).
  — When users may find it practical to, or prefer to, type information rather than selecting it from a list.
  — When users can recall and type information faster than selecting from a list.
  — When it is useful to provide the users with an occasional reminder of the choices available.
  — Where data must be entered that is not contained in the selection list.
  — Where screen space is limited.

Figure 7.76 Windows 3.1 Drop-down combo box, closed.

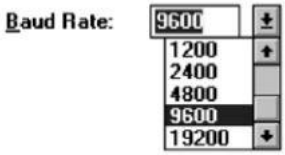
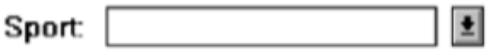


Figure 7.77 Windows 3.1 Drop-down combo box, opened.

  — For data and choices that are:
    - Best represented textually.
    - Somewhat familiar or known.
    - Ordered in an unpredictable fashion.
    - Frequently changed.
    - Large in number.
    - Variable or fixed in list length.

Prompt Button

- Provide a visual cue that a list box is hidden by including a downward-pointing arrow to the right of the text box.
- Separate the button from the text box by a small space.



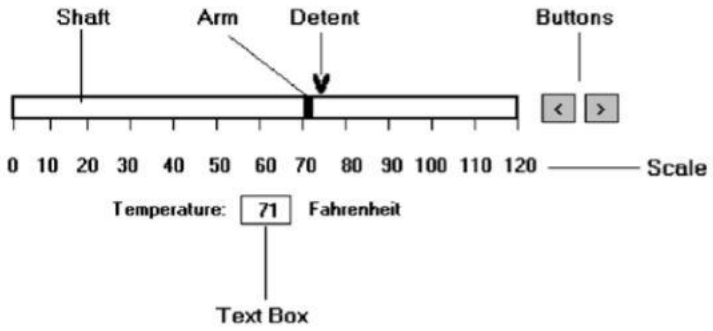
Other Guidelines

For the text box entry field, see the "Text Box/Single Line" guidelines. For the box and selection components, see the "Drop-down/Pop-up List Box guidelines".

OTHER OPERABLE CONTROLS

Slider

- Description:
  — A scale exhibiting degrees of a quality on a continuum.
  — Includes the following:
    - A shaft or bar.
    - A range of values with appropriate labels.
    - An arm indicating relative setting through its location on the shaft.
    - Optionally, a pair of buttons to permit incremental movement of the slider arm.
    - Optionally, a text box for typing or displaying an exact value.
    - Optionally, a detent position for special values.
      — May be oriented vertically or horizontally.
  — Selected by using the mouse to:
    - Drag a slider across the scale until the desired value is reached.
    - Point at the buttons at one end of the scale and clicking to change the value.
    - Keying a value in the associated text box.
- Purpose:
  — To make a setting when a continuous qualitative adjustment is acceptable, it is useful to see the current value relative to the range of possible values.
- Advantages:
  — Spatial representation of relative setting.
  — Visually distinctive.
- Disadvantages:
  — Not as precise as an alphanumeric indication.
  — Consumes screen space.
  — Usually more complex than other controls.
- Proper usage:
  — To set an attribute.
  — For mutually exclusive choices.
  — When an object has a limited range of possible settings.
  — When the range of values is continuous.
  — When graduations are relatively fine.
  — When the choices can increase or decrease in some well-known, predictable, and easily understood way.
  — When a spatial representation enhances comprehension and interpretation.
  — When using a slider provides sufficient accuracy.



General

• Use standard sliders whenever available.

Caption and Labels

- Caption:
  — Provide meaningful, clear, and consistent captions.
  — Display them using mixed-case letters.
  — Position the caption to the left of the box.
  • Alternately, it may be positioned left-justified above the slider.
- Labels:
  — Provide meaningful and descriptive labels to aid in interpreting the scale.

Scale

- Show a complete range of choices.
- Mark the low, intermediate, and high ends of the scale.
- Provide scale interval markings, where possible.
- Provide consistent increments.
- Permit the user to change the units of measure.
- If the precise value of a quantity represented is important, display the value set in an adjacent text box.

Slider Arm

• If the user cannot change the value shown in a slider, do not display a slider arm.

Slider Buttons

- Provide slider buttons to permit movement by the smallest increment.
- If the user cannot change the value shown in a slider, do not display slider buttons.

Detents

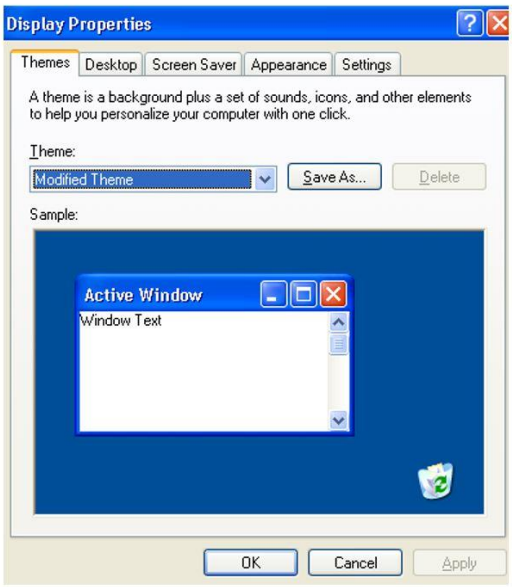
- Provide detents to set values that have special meaning.
- Permit the user to change the detent value.

Proportions

- To indicate the proportion of a value being displayed, fill the slider shaft in some visually distinctive way.
  — Fill horizontal sliders from left to right.
  — Fill vertical sliders from bottom to top.

Tabs

- Description: — A window containing tabbed dividers that create pages or sections. — Navigation is permitted between the pages or sections.
- Purpose: — To present information that can be logically organized into pages or sections within the same window.



- Advantages: — Resembles their paper-based cousins. — Visually distinctive. — Effectively organize repetitive, related information.
- Disadvantages: — Visually complex.
- Proper usage: — To present discrete, logically structured, related, information. — To present the setting choices that can be applied to an object. — When a short tab label can meaningfully describe the tab's contents. — When the order of information use varies.

Sections and Pages

- Place related information within a page or section.
- Order them meaningfully.
- Arrange pages so they appear to go deeper, left to right and top to bottom.
- Provide pages of equal size.

Location, Size, and Labels

- Place the tabs at the top of the page or section.
- Provide fixed-width tabs for pages or sections of related information.
- Provide textual labels.
  — Use system fonts.
  — Keep information brief and the same general length.
  • Nouns are usually better than verbs.
  — Use mixed case, capitalizing each significant word.
  — Assign a keyboard equivalent for keyboard access.
- Center the labels within the tabs.
- Restrict tabs to only one row.
- Arrange tabs so that they appear to go deeper, left to right and top to bottom.

Command Buttons

- If they affect only a page or section, locate the buttons on the page or section.
- If they affect the entire tabbed control, locate the buttons outside the tabbed pages

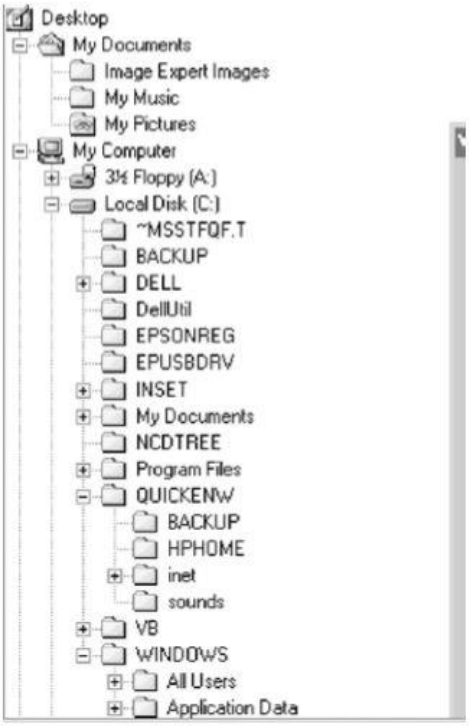
Date-Picker

- Description:
  — A drop-down list box that displays a 1-month calendar in the drop-down list box.
  — The displayed month can be changed through pressing command buttons with left- and right-pointing arrows.
    - The left arrow moves backward through the monthly calendars.
    - The right arrow moves forward through the monthly calendars.
  — A date for the list box can be selected from the drop-down calendar.
- Purpose:
  — To select a date for inscribing in a drop-down list box.
- Advantages:
  — Provides a representation of a physical calendar, a meaningful entity.
  — The calendar listing is ordered in a predictable way.
  — Visually distinctive.
- Disadvantages:
  — Requires an extra step to display the calendar.
  — When displayed, all month choices are not visible, requiring a form of scrolling to access the desired choice.

- Proper usage:
  — To select and display a single date in close monthly proximity to the default month presented on the drop-down list box.

Tree View

- Description:
  — A special list box control that displays a set of objects as an indented outline, based on the objects' logical hierarchical relationship.



— Includes, optionally, buttons that expand and collapse the outline.
  - A button inscribed with a plus (+) expands the outline.
  - A button inscribed with a minus (-) collapses the outline.

— Elements that can optionally be displayed are:
  - Icons.

  - Graphics, such as a check box.
  - Lines to illustrate hierarchical relationships.
- Purpose and proper usage:
  — To display a set of objects as an indented outline to illustrate their logical hierarchical relationship.

Scroll Bars

- Description:
  — An elongated rectangular container consisting of:
    - A scroll area.
    - A slider box or elevator inside.
    - Arrows or anchors at either end.
    - 
  — Available, if needed, in primary and secondary windows, some controls, and Web pages.
  — May be oriented vertically or horizontally at the window or page edge.
- Purpose:
  — To find and view information that takes more space than the allotted display space.
- Advantages:
  — Permits viewing data of unlimited size.
- Disadvantages:
  — Consumes screen space.
  — Can be cumbersome to operate.
- Proper use:
  — When more information is available than the window space for displaying it.
  — Do not use to set values.

Scroll Bar Design Guidelines

- General:
  — Provide a scroll bar when invisible information must be seen.
- Scroll area or container:
  — To indicate that scrolling is available, a scroll area or container should be provided.
  • Construct it of a filled-in bar displayed in a technique that visually contrasts with the window and screen body background.
- Scroll slider box or handle:
  — To indicate the location and amount of information being viewed, provide a slider box or handle.

  - Constructed of a movable and sizable open area of the scroll area, displayed in a technique that contrasts with the scroll area.
  - By its position, spatially indicate the relative location in the file of the information being viewed.
  - By its size, indicate, proportionately, the percentage of the available information in the file being viewed.
- Scroll directional arrows:
  — To indicate the direction in which scrolling may be performed, directional arrows should be provided.
  • Construct them as arrows in small boxes, with backgrounds that contrast with the scroll area.
- Selection:
  — When the slider box/handle has been selected, highlight it in some visually distinctive way.
- Location:
  — Position a vertical (top-to-bottom) scroll bar to the right of the window.
  — Position a horizontal (left-to-right) scroll bar at the bottom of the window.
- Size:
  — A vertical scroll bar should be the height of the scrollable portion of the window body.
  — A horizontal scroll bar should be at least one-half the width of the scrollable portion of the window body.
- Current state indication:
  — Whenever the window's size or the position of the information changes, the scroll bar components must also change, reflecting the current state.
  — Include scroll bars in all sizable windows.
  • If no information is currently available by scrolling in a particular direction, the relevant directional arrow should be subdued or grayed out.

Scroll Bar Usage Guidelines

- Scroll bar style:
  — Stick with standard, proven design styles.
- Directional preference:
  — Use vertical (top-to-bottom) scrolling.
  — Avoid horizontal (left-to-right) scrolling.

Media Controls

- For all playable files provide the following controls.
  — Play.
  — Pause/Resume.
  — Stop.

— Rewind.
— Fast Forward.
— Volume.

CUSTOM CONTROLS

- Implement custom controls with caution.
- If used, make the look and behaviour of custom controls different from that of standard controls.

PRESENTATION CONTROLS

Common presentation controls are static text fields, group boxes column headings, ToolTips, balloon tips, and progress indicators.

Static Text Fields

- Description:
  — Read-only textual information.
- Purpose:
  — To identify a control by displaying a control caption.
  — To clarify a screen by providing instructional or prompting information.
  — To present descriptive information.
- Proper usage:
  — To display a control caption.
  — To display instructional or prompting information.
  — To display descriptive information.

Static Text Field Guidelines

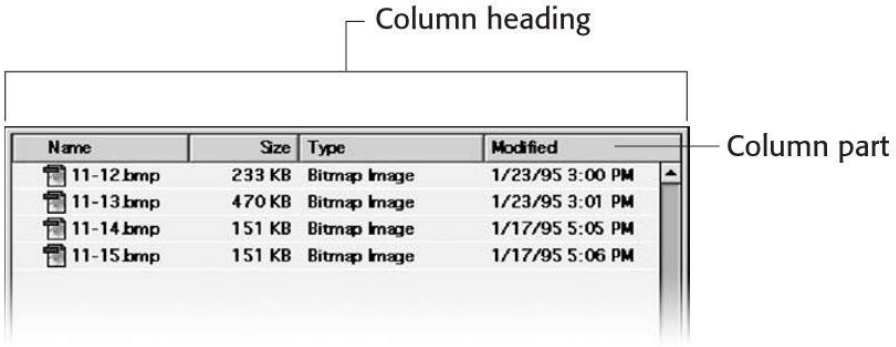
- Captions:
  — Include a colon (:) as part of the caption.
  — Include a mnemonic for keyboard access.
  — When the associated control is disabled, display it dimmed.
  — Follow all other presented guidelines for caption presentation and layout.
- Instructional or prompting information:
  — Display it in a unique and consistent font style for easy recognition and differentiation.
  — Follow all other presented guidelines for prompting and instructional information.
- Descriptive information:
  — Follow all other guidelines for required screen or control descriptive information.

Group Boxes

- Description:
  — A rectangular frame that surrounds a control or group of controls.
  — An optional caption may be included in the frame's upper-left corner.
- Purpose:
  — To visually relate the elements of a control.
  — To visually relate a group of related controls.
- Proper usage:
  — To provide a border around radio button or check box controls.
  — To provide a border around two or more functionally related controls.
- Guidelines:
  — Label or heading:
    - Typically, use a noun or noun phrase for the label or heading.
    - Provide a brief label or heading, preferably one or two words.
    - Relate label or heading's content to the group box's content.
    - Capitalize the first letter of each significant word.
    - Do not include and ending colon (:).
  — Follow all other guidelines presented for control and section borders.

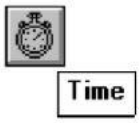
Column Headings

- Description: — Read-only textual information that serves as a heading above columns of text or numbers. — Can be divided into two or more parts.
- Purpose: — To identify a column of information contained in a table.
- Proper usage: — To display a heading above a column of information contained in a table.
- Guidelines:
  — Heading:
    - Provide a brief heading.
    - Can include text and a graphic image.
    - Capitalize the first letter of each significant word.
    - Do not include an ending colon (:).
  — The width of the column should fit the average size of the column entries.
  — Does not support keyboard access.



ToolTips

- Description:
  — A small pop-up window containing descriptive text that appears when a pointer is moved over a control or element either:
    - Not possessing a label.
    - In need of additional descriptive or status information.



- Purpose:
  — To provide descriptive information about a control or screen element.
- Advantages:
  — Identifies an otherwise unidentified control.
  — Reduces possible screen clutter caused by control captions and descriptive information.
  — Enables control size to be reduced.
- Disadvantages:
  — Not obvious, must be discovered.
  — Inadvertent appearance can be distracting.
- Proper usage:
  — To identify a control that has no caption.
  — To provide additional descriptive or status information about a screen element.

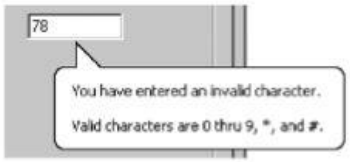
ToolTip Guidelines

- Display after a short time-out.
- For toolbars, provide a brief word as a label.
  — Use mixed case in the headline style of presentation with no ending punctuation.
- For other elements, provide a brief phrase presenting descriptive or status information.
  — Use mixed case in the sentence style of presentation.
- Present ToolTips at the lower-right edge of the pointer.
  — Display them fully on the screen.

— For text boxes, display ToolTips centered under the control.
- Display them in the standard system ToolTip colours.
- Remove the ToolTip when the control is activated or the pointer is moved away.
- Don't substitute ToolTips for good design.

Balloon Tips

- Description:
  — A small pop-up window that contains information in a word balloon.
  — Components can include:
    - Title.
    - Body text.
    - Message Icons.
  — Appear adjacent to the item to which they apply, generally above or to left.
  — Only one tip, the last posted, is visible at any time.
  — Tips are removed after a specified time period.
- Purpose:
  — To provide additional descriptive or status information about a screen element.
- Advantages:
  — Provides useful reminder and status information.
- Disadvantages:
  — If overused they lose their attention-getting value.
  — If overused in situations the user considers not very important, their continual appearance can be aggravating.
- Proper usage:
  — To display noncritical:
    - Reminder information.
    - Notification information.
  — Do not use tips to display critical information.



Balloon Tip Guidelines

- General:
  — Use a notification tip to inform the user about state changes.
  — Use a reminder tip for state changes that the user might not usually notice.
  — Point the tip of the balloon to the item it references.
  — Do not use them to replace ToolTips.

— Do not overuse balloon tips.

- Content:
  — Restrict them to a length of 100 characters, including title and body text.
  — Title text should:
    - If the tip refers to an icon or other image representing a specific object, include:
      — The object's name, using its normal capitalization.
      — The object's status, using sentence-style presentation without ending punctuation.
    - Be presented in bold.

  — Body text should:
    - Include a description of the situation in one or two brief sentences.
    - Include a brief suggestion for correcting the situation.
    - Be presented using mixed-case in the sentence style.

Progress Indicators

- Description:
  — A rectangular bar that fills as a process is being performed, indicating the percentage of the process that has been completed.
- Purpose:
  — To provide feedback concerning the completion of a lengthy operation.
- Proper usage:
  — To provide an indication of the proportion of a process completed.

Progress Indicator Guidelines

- When filling the indicator:
  — If horizontally arrayed, fill it from left to right.
  — If vertically arrayed, fill it from bottom to top.
- Fill it with a colour or a shade of gray.
- Include descriptive text for the process, as necessary.
- Place text outside of the control.

Sample Box

- Description:
  — A box illustrating what will show up on the screen based upon the parameter or parameters selected.
  — May include text, graphics, or both.
- Purpose:
  — To provide a representation of actual screen content based upon the parameter or parameters selected.
- Guidelines:
  — Include a brief label.
  — Use mixed case in the headline style.
  — Locate it adjacent to the controls upon which it is dependent.

Scrolling Tickers

- Description:
  — Text that scrolls horizontally through a container window.
- Advantages:
  — Consume less screen space than full text.
- Disadvantages:
  — Hard to read.
  — Time-consuming to interpret.
  — Distracting.
- Guideline:
  — Do not use.

SELECTING THE PROPER CONTROLS

The proper control will enable a person to make needed selections, entries, and changes quickly, efficiently, and with fewer mistakes. Improper selection most often leads to the opposite result.

Entry versus Selection—A Comparison

Studies looked at the advantages and disadvantages of using either entry fields or selection fields for data collection. Entry involved keying text; selection was performed by pointing at a choice through the keyboard using the cursor control keys (not a mouse). The information compared was of three kinds: dates, text, and data. The first conclusion:

Choosing a Type of Control

• For familiar, meaningful data choose the technique that, in theory, requires the fewest number of keystrokes to complete.

• If the data is unfamiliar or prone to typing errors, choose a selection technique

Aided versus Unaided Entry •

- Provide aided entry whenever possible.
  — Absorb any extra and unnecessary keystrokes.
  — Provide an auditory signal that auto completion has been performed.

Comparison of GUI Controls

Table 7.1 Controls Evaluated by Tullis and Kodimer (1992)

DIRECT MANIPULATION
1. Drag and Drop On • The items are arrayed horizontally. An item is dragged to a new location above another item and released. The item in that position moves to the old location of the arriving item.
2. Drag and Drop Between • The items are arrayed horizontally. An item is dragged to a new location between two other items and released. The items are readjusted into new positions, including, when necessary, automatic wrap-around for items located at the end of the line.
SELECTION
3. Icons • The items are arrayed horizontally. Icons are positioned between each pair of items. Selecting an icon switches the positions of each adjacent item.
4. Radio Buttons • The items are presented in a matrix, item name along the left side, item position numbers across the top. Radio buttons in the matrix are selected to represent each item's position.
5. Menus (Drop-down List Boxes) • Items are positioned horizontally. A drop-down listing is activated, and the item for that location selected.
ENTRY
6. One Entry Area • A single text entry field is provided. A one-character mnemonic (F,N,S,D) is provided for each choice. The mnemonics are keyed in the order in which the items are to be arrayed.
7. Four Entry Areas • Four text entry fields, labeled with the item names are arranged vertically. A number (1-4) is keyed into each field, indicating the manner in which the items are to be ordered.

Mutually Exclusive Choice Controls

- For a small set of options (5), a medium set (12), and a large set (30), radio buttons were significantly faster than the other mutually exclusive controls.
- The medium and large set sizes (12 and 30) are larger than generally recommended for radio buttons (8 or less). The results indicate that radio buttons may effectively be used for these larger quantities

Nonexclusive Choice Controls

• For a small set of options (5) with two selected choices, a medium set (12) with three selected choices, and a large set (30) with eight selected choices, check boxes were significantly faster than the other nonexclusive controls.

Controls for Selecting a Value within a Range

- Setting range values included indicating a time, a percentage, or the transmission frequency of a radio station.
- Controls evaluated were the spin button, text entry field, and the slider.
- The spin button was the most accurate, and the text entry field fastest and most preferred.
- General conclusions are:
    - Making all options always visible will enhance performance.
    - Requiring additional actions to make further options visible slows performance.
    - For longer lists, scrolling tends to degrade performance more than the action associated with retrieving a hidden list.

Control Selection Criteria

- Selection of the proper control, then, depends on several factors. The first is the structure and characteristics of the property or data itself.
- Other considerations include the nature of the task, the nature of the user, and the limitations of the display screen itself
- Data considerations include the following:
    - Is the property or data mutually exclusive or nonexclusive? Does entry/selection require single or multiple items?
    - Is the property or data discrete or continuous? Discrete data can be meaningfully specified and categorized, while continuous data cannot.
    - Is the property or data limited or unlimited in scope? If limited, how many items will the data normally not exceed?
    - Is the property or data fixed or variable in list length? Are there always a fixed number of items, or will it varies?
    - Is the property or data ordered in a predictable or unpredictable fashion? If predictable, will the user be able to anticipate the next, unseen, item?
    - Can the property or data be represented pictorially? Will a picture or graphic be as meaningful as a textual description?
- Task considerations reflect the nature of the job. Considerations include the following:
    - How often is an item entered or selected?
    - How often is an item changed?
    - How precisely must the item be entered or selected?
- User considerations reflect the characteristics of the user. Important considerations:

  - How much training in cotrol operation will be provided?
  - How meaningful or known is the property or data to the user?
  - How easily remembered or learned by the user is the property or data?
  - How frequently used will the system be?
  - Is the user an experienced typist?
- Display considerations reflect the characteristics of the screen and hardware.
  - How much screen space is available to display the various controls?

Choosing a Control Form

When to Permit Text Entry

- Permit text entry if any of the following questions can be answered Yes:
  — Is the data unlimited in size and scope?
  — Is the data familiar?
  — Is the data not conducive to typing errors?
  — Will typing be faster than choice selection?
  — Is the user an experienced typist?

What Kind of Control to Choose

Table 7.2 Best Controls for Certain Tasks and Screen Conditions

TASK	BEST CONTROL	IF SCREEN SPACE CONSTRAINTS EXIST
Mutually Exclusive	Radio Buttons	Drop-down/Pop-up List Box
Not Mutually Exclusive	Check Boxes	Multiple-Selection List Box
Select or Type a Value Text Entry Field	Radio Buttons with "Other"	Drop-down Combo Box
Setting a Value within a Range	Spin Button	Text Box

From Johnsgard et al., 1995

Table 7.3 Suggested Uses for Graphical Controls

1. IF:	*USE:*
• *Mutually exclusive* alternatives. • Discrete data. • Best represented verbally. • Very limited in number (2 to 8).	
AND: • Typed entry is never necessary. • Content can never change. • Adequate screen space is available.	Radio Buttons
OR: • Typed entry is never necessary. • Content can never change. • Adequate screen space is not available.	Drop-down/Pop-up List Box
OR: • Typed entry may be necessary. • Content can change. • Adequate screen space is available.	Combo box
OR: • Typed entry may be necessary. • Content can change. • Adequate screen space is not available.	Drop-down/Pop-up Combo Box
2. IF:	*USE:*
• *Mutually exclusive* alternatives. • Discrete data. • Best represented verbally. • Potentially large in number (9 or more).	
AND: • Typed entry is never necessary. • Content can never change. • Adequate screen space is available.	Single-Selection List Box
OR: • Typed entry is never necessary. • Content can never change. • Adequate screen space is not available.	Drop-down/Pop-up List Box
OR: • Typed entry may be necessary. • Content can change. • Adequate screen space is available.	Combo Box
OR: • Typed entry may be necessary. • Content can change. • Adequate screen space is not available.	Drop-down/Pop-up Combo Box

Next are two tables providing some control recommendations based upon a control's known advantages, disadvantages, and proper usage characteristics

Table 7.3 Continued

3. IF:	*USE:*
• *Mutually exclusive* alternatives. • Discrete data. • Best represented graphically. • Content rarely changes. • Small or large number of items.	Palette
4. IF:	*USE:*
• *Mutually exclusive* alternatives. • Not frequently selected. • Content does not change. • Well-known, easily remembered data. • Predictable, consecutive data. • Typed entry sometimes desirable.	
AND: • Adequate screen space is not available.	Spin Box
OR: • Adequate screen space is available.	Combo Box
5. IF:	*USE:*
• *Mutually exclusive* alternatives. • Continuous data with a limited range of settings. • Value increases/decreases in a well-known, predictable way. • Spatial representation enhances comprehension.	Slider
6. IF:	*USE:*
• *Nonexclusive* alternatives. • Discrete data. • Best represented verbally. • Typed entry is never necessary. • Content can never change. • Adequate screen space is available.	
AND: • Very limited in number (2 to 8).	Check Boxes
OR: • Potentially large in number (9 or more).	Multiple-Selection List Box

Table 7.4 Choosing a Command Form

IF THE COMMANDS:	USE:
Are standard commands provided by a tool set.	Commands provided by the tool set
Total seven or more, and can be arranged hierarchically into groups.	Menu bar and pull-downs
Total six or fewer, are selected frequently, and affect an entire window.	Buttons in a window
Total seven or more, are selected frequently, and affect an entire window.	Buttons in a toolbar
Are used with other controls, or are complicated commands and need to be simplified.	Buttons in a dialog box
Are sometimes used frequently and are sometimes used infrequently.	Buttons in a dialog box
Are frequently accessed and have only two conditions.	Toggled menu item

Choosing between Buttons and Menus for Commands

Determining the proper way to present a command also depends on several factors. The following considerations are involved in choosing the correct command form:

- Whether or not the command part of a standard tool set.
- The total number of commands in the application.
- The complexity of the commands.
- The frequency with which commands are used. Whether
- or not the command is used in association with another control.

CHAPTER 10. SELECT THE PROPER KIND OF WINDOWS

Test, Test, and Retest

Testing steps to be reviewed are:

- Identifying the purpose and scope of testing.
- Understanding the importance of testing.
- Developing a prototype.
- Developing the right kind of test plan.
- Designing a test to yield relevant data.
- Soliciting, selecting, and scheduling users to participate.
- Providing the proper test facility.
- Conducting tests and collecting data.
- Analyzing the data and generating design recommendations.
- Modifying the prototype as necessary.
- Testing the system again.
- Evaluating the working system.

The Purpose of Usability Testing

- First, it establishes a communication bridge between developers and users. Through testing, the developer learns about the user's goals, perceptions, questions, and problems.
- Second, testing is used to evaluate a product. It validates design decisions. It also can identify potential problems in design at a point in the development process where they can be more easily addressed.

The Importance of Usability Testing

A thorough usability testing process is important for many reasons,

- Developers and users possess different models.
- Developer's intuitions are not always correct.
- There is no average user.
- It's impossible to predict usability from appearance.
- Design standards and guidelines are not sufficient.
- Informal feedback is inadequate.
- Problems found late are more difficult and expensive to fix.
- Advantages over a competitive product can be achieved.

Scope of Testing

- Testing should begin in the earliest stages of product development and continue throughout the development process.
- It should include as many of the user's tasks, and as many of the product's components, as reasonably possible.

Prototypes

- A prototype is primarily a vehicle for exploration, communication, and evaluation. Its purpose is to obtain user input in design, and to provide feedback to designers.
- A prototype is a simulation of an actual system that can be quickly created.
- A prototype may be a rough approximation, such as a simple hand-drawn sketch, or it may be interactive, allowing the user to key or select data using controls, navigate through menus, retrieve displays of data, and perform basic system functions.
- A prototype may have great breadth, including as many features as possible to present concepts and overall organization, or it might have more depth, including more detail on a given feature or task to focus on individual design aspects.
- Various Kinds of Prototypes in general order of increased fidelity are as follows

1. Hand Sketches and Scenarios

- Description:
  — Screen sketches created by hand.
  — Focus is on the design, not the interface mechanics.
  — A low-fidelity prototype.
- Advantages:
  — Can be used very early in the development process.
  — Suited for use by entire design team.
  — No large investment of time and cost.
  — No programming skill needed.
  — Easily portable.
  — Fast to modify and iterate.
  — A rough approximation often yields more substantive critical comments.

— Easier to comprehend than functional specifications.
— Can be used to define requirements.

- Disadvantages:
  — Only a rough approximation.
  — Limited in providing an understanding of navigation and flow.
  — A demonstration, not an exercise.
  — Driven by a facilitator, not the user.
  — Limited usefulness for a usability test.
  — A poor detailed specification for writing the code.
  — Usually restricted to most common tasks.

2. Sketch Creation Process

- Sketch (storyboard) the screens while determining:
  — The source of the screen's information.
  — The content and structure of individual screens.
  — The overall order of screens and windows.
- Use an erasable medium.
- Sketch the screens needed to complete each workflow task.
- Try out selected metaphors and change them as necessary.
- First, storyboard common/critical/frequent scenarios.
  — Follow them from beginning to end.
  — Then, go back and build in exceptions.
- Don't get too detailed; exact control positioning is not important, just overall order and flow.
- Storyboard as a team, including at least one user.
- Only develop online prototypes when everyone agrees that a complete set of screens has been satisfactorily sketched.

3. Interactive Paper Prototypes

- Description:
  — Interface components (menus, windows, and screens) constructed of common paper technologies (Post-It notes, transparencies, and so on).
  — The components are manually manipulated to reflect the dynamics of the software.
  — A low-fidelity prototype.
- Advantages:
  — More illustrative of program dynamics than sketches.
  — Can be used to demonstrate the interaction.
  — Otherwise, generally the same as for hand-drawn sketches and scenarios.
- Disadvantages:
  — Only a rough approximation.

— A demonstration, not an exercise.
— Driven by a facilitator, not the user.
— Limited usefulness for usability testing.

4. **Programmed Facades**
   - Description:
     — Examples of finished dialogs and screens for some important aspects of the system.
     — Created by prototyping tools.
     — Medium-fidelity to high-fidelity prototypes.
   - Advantages:
     — Provide a good detailed specification for writing code.
     — A vehicle for data collection.
   - Disadvantages:
     — May solidify the design too soon.
     — May create the false expectation that the "real thing" is only a short time away.
     — More expensive to develop.
     — More time-consuming to create.
     — Not effective for requirements gathering.
     — Not all of the functions demonstrated may be used because of cost, schedule limitations, or lack of user interest.
     — Not practical for investigating more than two or three approaches.

5. **Prototype-Oriented Languages**
   - Description:
     — An example of finished dialogs and screens for some important aspects of the system.
     — Created through programming languages that support the actual programming process.
     — A high-fidelity prototype.
   - Advantages:
     — May include the final code.
     — Otherwise, generally the same as those of programmed facades.
   - Disadvantages:
     — Generally the same as for programmed facades.

KINDS OF TESTS

A test is a tool that is used to measure something. The "something" may be:

- Conformance with a requirement.
- Conformance with guidelines for good design.

- Identification of design problems.
- Ease of system learning.
- Retention of learning over time.
- Speed of task completion.
- Speed of need fulfillment.
- Error rates.
- Subjective user satisfaction.

1. **Guidelines Review**
   - Description:
     — A review of the interface in terms of an organization's standards and design guidelines.
   - Advantages:
     — Can be performed by developers.
     — Low cost.
     — Can identify general and recurring problems
     — Particularly useful for identifying screen design and layout problems.
   - Disadvantages:
     — May miss severe conceptual, navigation, and operational problems.

2. **Heuristic Evaluation**
   - Description:
     — A detailed evaluation of a system by interface design specialists to identify problems.
   - Advantages:
     — Easy to do.
     — Relatively low cost.
     — Does not waste user's time.
     — Can identify many problems.
   - Disadvantages:
     — Evaluators must possess interface design expertise.
     — Evaluators may not possess an adequate understanding of the tasks and user communities.
     — Difficult to identify system wide structural problems.
     — Difficult to uncover missing exits and interface elements.
     — Difficult to identify the most important problems among all problems uncovered.
     — Does not provide any systematic way to generate solutions to the problems uncovered.
   - Guidelines:
     — Use 3 to 5 expert evaluators.
     — Choose knowledgeable people:

• Familiar with the project situation.
• Possessing a long-term relationship with the organization.

Heuristic Evaluation Process

- Preparing the session:
  — Select evaluators.
  — Prepare or assemble:
    - A project overview.
    - A checklist of heuristics.
  — Provide briefing to evaluators to:
    - Review the purpose of the evaluation session.
    - Preview the evaluation process.
    - Present the project overview and heuristics.
    - Answer any evaluator questions.
    - Provide any special evaluator training that may be necessary.
- Conducting the session:
  — Have each evaluator review the system alone.
  — The evaluator should:
    - Establish own process or method of reviewing the system.
      — provide usage scenarios, if necessary.
    - Compare his or her findings with the list of usability principles.
    - Identify any other relevant problems or issues.
    - Make at least two passes through the system.
  — Detected problems should be related to the specific heuristics they violate.
  — Comments are recorded either:
    - By the evaluator.
    - By an observer.
  — The observer may answer questions and provide hints.
  — Restrict the length of the session to no more than 2 hours.
- After the session:
  — Hold a debriefing session including observers and design team members where:
    - Each evaluator presents problems detected and the heuristic it violated.
    - A composite problem listing is assembled.
    - Design suggestions for improving the problematic aspects of the system are discussed.
  — After the debriefing session:
    - Generate a composite list of violations as a ratings form.
    - Request evaluators to assign severity ratings to each violation.
    - Analyze results and establish a program to correct violations and deficiencies.

Heuristic Evaluation Effectiveness

One of the earliest papers addressing the effectiveness of heuristic evaluations was by Nielsen (1992). He reported that the probability of finding a major usability problem averaged 42 percent for single evaluators in six case studies. The corresponding probability for uncovering a minor problem was only 32 percent.

Heuristic evaluations are useful in identifying many usability problems and should be part of the testing arsenal. Performing this kind of evaluation before beginning actual testing with users will eliminate a number of design problems, and is but one step along the path toward a very usable system.

Research based set of heuristics

Table 14.1 Severity Ratings in Heuristic Evaluation

0 = I don't agree that this is a usability problem at all.
1 = A cosmetic problem only. Need not be fixed unless extra time is available.
2 = A minor usability problem. Fixing should be given a low priority.
3 = A major usability problem. Important to fix and should be given a high priority.
4 = A usability catastrophe. Imperative to fix before the product can be released.

From useit.com

Table 14.2 Research-Based Set of Heuristics

1. Automate unwanted workload. • Free cognitive resources for high-level tasks. • Eliminate mental calculations, estimations, comparisons, and unnecessary thinking.
2. Reduce uncertainty. • Display data in a manner that is clear and obvious.
3. Fuse data. • Reduce cognitive load by bringing together lower-level data into a higher-level summation.
4. Present new information with meaningful aids to interpretation. • Use a familiar framework, making it easier to absorb. • Use everyday terms, metaphors, and so on.
5. Use names that are conceptually related to functions. • Context-dependent. • Attempt to improve recall and recognition.
6. Group data in consistently meaningful ways to decrease search time.
7. Limit data-driven tasks. • Reduce the time needed to assimilate raw data. • Make appropriate use of color and graphics.
8. Include in the displays only that information needed by a user at a given time. • Allow users to remain focused on critical data. • Exclude extraneous information that is not relevant to current tasks.
9. Provide multiple coding of data where appropriate.
10. Practice judicious redundancy. • To resolve the conflict between heuristics 6 and 8.

From Gerhardt-Powals (1996).

Table 14.3 Possible Web Page Heuristics

1. Speak the user's language. • Use familiar words, phrases, and concepts. • Present information in a logical and natural order.
2. Be consistent. • Indicate similar concepts through identical terminology and graphics. • Adhere to uniform conventions for layout, formatting, typefaces, labeling, and so on.
3. Minimize the user's memory load. • Take advantage of recognition rather than recall. • Do not force users to remember key information across documents.
4. Build flexible and efficient systems. • Accommodate a range of user sophistication and diverse user goals. • Provide instructions where useful. • Lay out screens so that frequently accessed information is easily found.

(*continues*)

Table 14.3 Continued

5. Design aesthetic and minimalist systems. • Create visually pleasing displays. • Eliminate information that is irrelevant or distracting.
6. Use chunking. • Write materials so that documents are short and contain only one topic. • Do not force the user to access multiple documents to complete a single thought.
7. Provide progressive levels of detail. • Organize information hierarchically, with more general information appearing before more specific detail. • Encourage the user to delve as deeply as needed, but to stop whenever sufficient information has been obtained.
8. Give navigational feedback. • Facilitate jumping between related topics. • Allow the user to determine his/her current position in the document structure. • Make it easy to return to an initial state.
9. Don't lie to the user. • Eliminate erroneous or misleading links. • Do not refer to missing information.

From Levi and Conrad (1996).

3. Cognitive Walkthroughs

- Description:
  — Reviews of the interface in the context of tasks users perform.
- Advantages:
  — Allow a clear evaluation of the task flow early in the design process.
  — Do not require a functioning prototype.
  — Low cost.
  — Can be used to evaluate alternate solutions.
  — Can be performed by developers.
  — More structured than a heuristic evaluation.
  — Useful for assessing exploratory learning.
- Disadvantages:
  — Tedious to perform.
  — May miss inconsistencies and general and recurring problems.
- Guidelines:
  — Needed to conduct the walkthrough are:
    - A general description of proposed system users and what relevant knowledge they possess.
    - A specific description of one or more core or representative tasks to be performed.
    - A list of the correct actions required to complete each of the tasks.
  — Review:

- Several core or representative tasks across a range of functions.
- Proposed tasks of particular concern.

— Developers must be assigned roles of:
- Scribe to record results of the action.
- Facilitator to keep the evaluation moving.

— Start with simple tasks.
— Don't get bogged down demanding solutions.
— Limit session to 60 to 90 minutes.

4. Think-Aloud Evaluations

- Description:
  — Users perform specific tasks while thinking out load.
  — Comments are recorded and analyzed.
- Advantages:
  — Utilizes actual representative tasks.
  — Provides insights into the user's reasoning.
- Disadvantages:
  — May be difficult to get users to think out loud.
- Guidelines:
  — Develop:
  - Several core or representative tasks.
  - Tasks of particular concern.

  — Limit session to 60 to 90 minutes.

5. Usability Test

- Description:
  — An interface evaluation under real-world or controlled conditions.
  — Measures of performance are derived for specific tasks.
  — Problems are identified.
- Advantages:
  — Utilizes an actual work environment.
  — Identifies serious or recurring problems.
- Disadvantages:
  — High cost for establishing facility.
  — Requires a test conductor with user interface expertise.
  — Emphasizes first-time system usage.
  — Poorly suited for detecting inconsistency problems.

6. **Classic Experiments**
   - Description:
     — An objective comparison of two or more prototypes identical in all aspects except for one design issue.
   - Advantages:
     — Objective measures of performance are obtained.
     — Subjective measures of user satisfaction may be obtained.
   - Disadvantages:
     — Requires a rigorously controlled experiment to conduct the evaluation.
     — The experiment conductor must have expertise in setting up, running, and 192nalysing the data collected.
     — Requires creation of multiple prototypes.
   - Guidelines:
     — State a clear and testable hypothesis.
     — Specify a small number of independent variables to be manipulated.
     — Carefully choose the measurements.
     — Judiciously select study participants and carefully or randomly assign them to groups.
     — Control for biasing factors.
     — Collect the data in a controlled environment.
     — Apply statistical methods to data analysis.
     — Resolve the problem that led to conducting the experiment.
7. **Focus Groups**
   - Description:
     — A discussion with users about interface design prototypes or tasks.
   - Advantages:
     — Useful for:
       - Obtaining initial user thoughts.
       - Trying out ideas.
     — Easy to set up and run.
     — Low cost.
   - Disadvantages:
     — Requires experienced moderator.
     — Not useful for establishing:
       - How people really work.
       - What kinds of usability problems people have.
   - Guidelines:
     — Restrict group size to 8 to 12.
     — Limit to 90 to 120 minutes in length.
     — Record session for later detailed analysis.

Choosing a Testing Method

- Beer, Anodenko, and Sears (1997) suggest a good pairing is cognitive walkthroughs followed by think-aloud evaluations.
- Using cognitive walkthroughs early in the development process permits the identification and correction of the most serious problems. Later, when a functioning prototype is available, the remaining problems can be identified using a think-aloud evaluation.
- A substantial leap forward in the testing process would be the creation of a software tool simulating the behavior of people. This will allow usability tests to be performed without requiring real users to perform the necessary tasks.
- In conclusion, each testing method has strengths and weaknesses. A well-rounded testing program will use a combination of some, or all, of these methods to guarantee the usability of its created product.
- It is very important that testing start as early as possible in the design process and, continue through all developmental stages.

Developing and Conducting the Test

A usability test requires developing a test plan, selecting test participants, conducting the test, and analyzing the test results.

The Test Plan

- Define the scope of the test.
  - A test's scope will be influenced by a variety of factors.
  - Determinants include the following issues:
  - The design stage: early, middle, or late—the stage of design influences the kinds of prototypes that may exist for the test,
  - the time available for the test—this may range from just a few days to a year or more,
  - finances allocated for testing—money allocated may range from one percent of a project's cost to more than 10 percent,
  - the project's novelty (well defined or exploratory)—this will influence the kinds of tests feasible to conduct,
  - expected user numbers (few or many) and interface criticality (life-critical medical system or informational exhibit)—much more testing depth and length will be needed for systems with greater human impact, and finally, the development team's experience and testing knowledge will also affect the kinds of tests that can be conducted.
- Define the purpose of the test.
  — Performance goals.
  — What the test is intended to accomplish.
- Define the test methodology.

— Type of test to be performed.
— Test limitations.
— Developer participants.

- Identify and schedule the test facility or location.
    - The location should be away from distractions and disturbances. If the test is being held in a usability laboratory, the test facility should resemble the location where the system will be used.
    - It may be an actual office designated for the purpose of testing, or it may be a laboratory specially designed and fitted for conducting tests.
- Develop scenarios to satisfy the test's purpose.

Test Participants

Assemble the proper people to participate in the test.

Test Conduct and Data Collection

To collect usable data, the test should begin only after the proper preparation. Then, the data must be properly and accurately recorded. Finally, the test must be concluded and followed up properly.

Usability Test Guidelines

- Before starting the test:
  — Explain that the objective is to test the software, not the participants.
  — Explain how the test materials and records will be used.
  — If a consent agreement is to be signed, explain all information on it.
  — If verbal protocols will be collected, let participants practice thinking aloud.
  — Ensure that all participants' questions are answered and that participants are comfortable with all procedures.
- During the test:
  — Minimize the number of people who will interact with the participants.
  — If observers will be in the room, limit them to two or three.
  — Provide a checklist for recording:
    - Times to perform tasks.
    - Errors made in performing tasks.
    - Unexpected user actions.
    - System features used/not used.
    - Difficult/easy-to-use features.
    - System bugs or failures.

  — Record techniques and search patterns that participants employ when attempting to work through a difficulty.
  — If participants are thinking aloud, record assumptions and inferences being made.
  — Record the session with a tape recorder or video camera.

— Do not interrupt participants unless absolutely necessary.
— If participants need help, provide some response.
  - Provide encouragement or hints.
  - Give general hints before specific hints.
  - Record the number of hints given.
— Watch carefully for signs of stress in participants:
  - Sitting for long times doing nothing.
  - Blaming themselves for problems.
  - Flipping through documentation without really reading it.
— Provide short breaks when needed.
— Maintain a positive attitude, no matter what happens.

- After the test:
  — Hold a final interview with participants; tell participants what has been learned in the test.
  — Provide a follow-up questionnaire that asks participants to evaluate the product or tasks performed.
  — If videotaping, use tapes only in proper ways.
  • Respect participants' privacy.
  • Get written permission to use tapes.

Analyze, Modify, and Retest

- ✓ Compile the data from all test participants.
- ✓ List the problems the participants had.
- ✓ Sort the problems by priority and frequency.
- ✓ Develop solutions for the problems.
- ✓ Modify the prototype as necessary.
- ✓ Test the system again, and again.

Evaluate the Working System

- Collect information on actual system usage through:
  — Interviews and focus group discussions.
  — Surveys.
  — Support line.
  — Online suggestion box or trouble reporting.
  — Online bulletin board.
  — User newsletters and conferences.
  — User performance data logging.
- Respond to users who provide feedback.