USER INTERFACE DESIGN (18CS734)

SEMESTER – VII

Module I

The User Interface-Introduction, Overview, The importance of user interface – Defining the user interface, the importance of Good design, Characteristics of graphical and web user interfaces, Principles of user interface design

Chapter 1: Introduction

User Interface Design Definition

- 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.
- 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 people communicate his needs to the system using keyboard or any pointing device and output is how the system returns processed result to user through screen or sound.
- 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.

GUI Definition

- 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 (i.e. different elements referred to as objects). 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.

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.

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 allowedto 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.

Concept of 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.

Graphical system 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. 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**: symbolic displays are more natural and advantageous because the human mind has a powerful image memory.
- **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
- **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
- **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**: 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.

Graphical system 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.
- **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.
- **Window manipulation requirements**: Window handling and manipulation times are still excessive and repetitive. This wastes time
- **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.
- **Not always the preferred style of interaction**: Not all users prefer a pure iconic interface. User will also prefer alternatives with textual captions.
- **Not always fastest style of interaction:** graphic instructions on an automated bank teller machine were inferior to textual instructions.
- **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.

Chapter 2: Characteristics of the Graphical User Interface

It includes a set of seven defining concepts. They are -

1. **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.

2. **Pick-and-Click Interaction**
   - 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.

3. **Restricted Set of Interface Options**
   - The array of alternatives available to the user is what is presented on the screen orwhat may be retrieved through what is presented on the screen, nothing less, and nothing more. This concept fostered the acronym WYSIWYG.

4. **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 produceone that conveys the most relevant information. Effective visualizations can facilitate mental insights, increase productivity, and foster faster and more accurate use of data.

5. **Object Orientation**
   - A graphical system consists of objects and actions. Objects are what people see onthe 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.
   - IBM's SAA CUAbreaks Objects into three meaningful classes as Data objects, which present information, Container objects to hold other objects and Device objects represent physical objects in the real world.
   - Objects can exist within the context of other objects, and one object may affectthe way another object appears or behaves. These relationships are called collections, constraints, composites, and containers.
   - **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).
- 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** 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).

- **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.

6. Use of (a person's) Recognition Memory

- Continuous visibility of objects and actions encourages to eliminate ―out of sight, out of mind‖ problem

7. Concurrent Performance of Functions

- Graphic systems may do two or more things at one time. Multiple programs mayrun simultaneously.
- It may process background tasks (cooperative multitasking) or 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 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 others with a more agreeable interface.

Characteristics of Web Design

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

GUI v/s Web page Design: Both 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. Significant differences do exist however. The following table shows a summary listing of significant differences.

GUI versus Web Design

Characteristics	GUI	WEB
1. Devices	User hardware variations limited. User hardware characteristics well defined. Screens appear exactly as specified.	User hardware variations enormous. Screen appearanceinfluenced by hardware being used.
2. User Focus	Data and applications.	Information and navigation.
3. Data/ Information	Typically created and used by known and trusted Sources are trusted. Properties generally known.Typically placed intosystem by users or known people and organizations. Typically organized in a meaningful fashion. A notion of private & 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.
4. User Tasks	Install, configure, personalize, start, use, andOpen, use, and close data files. Familiarity with applications often achieved.	Link to a site, browse or read pages, fill out forms, upgrade programs. register for services, participate in transactions, download and save things. Familiarity with many sites not established.
5. User's Conceptual Space	Controlled & Constrained by program.	Infinite and generally unorganized.
6. Presentation Elements	Windows, menus, controls, data, toolbars, messages and so on. 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.
7. Navigation	Through menus, lists, trees, dialogs, and wizards. Not a strong and visible concept. Constrained by design.	Through links, bookmarks, and typed URLs. Significant and highly visible concept. Few standards.

8. 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.
9. 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.
10. Response Time	Nearly instantaneous	Quite variable, depending on transmission speeds, page content, and so on. Long times can upset the user.
11. 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 significant personalization.
12. 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.
13. 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.
14. Consistency	Major objective exists within and across applications. Aided by platform toolkit and design guidelines. Universal consistency inGUI 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.

15. User Assistance	Integral part of most systems and applications. Accessed through standard mechanisms. Documentation, both onlineand 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.
16. Integration	Seamless integration of all applications into the platform environment is 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.
17. 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.
18. Reliability	Tightly controlled inbusiness 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 becausetheir 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 toit. 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.
- **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.
- **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 andfooters on each Web page.

Merging Graphical business system and 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.
- **Tasks:** An intranet is used for an organization's everyday activities, including complex transactions, queries, and communications. The Internet is mainly usedto 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 toone 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 GUIapplications 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

- 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. It is a workstation, released in April-1981. Xerox star 8010 information system was first commercial computer with a GUI. It was composed of advances in hardware & software technology. It used a technique called "bit-mapping" in which everything on computer screen was in effect – a picture.

Principles for the Xerox STAR

- 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,
- 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 color 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 behavior 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.
- **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.
  - When an error does occur, provide constructive messages.

- **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.

USER INTERFACE DESIGN (18CS734)

SEMESTER – VII

Module – II

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.

The User Interface Design Process

Obstacles and Pitfalls in the Development Path

- Developing a computer system is never easy. 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 forchange.
  - Even if you have made the best system humanly possible, people will stillmake mistakes when using it.
  - Designers need good tools.
  - You must have behavioral 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 behavior 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. While obstacles to design will always exist, pitfalls can be eliminated if the following design commandments remain foremost in the designer's mind.

- **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.
- **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.

- **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.
- **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.
- **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 existfor design trade-offs to be thought out more carefully.

Usability

- The term usability used to describe the effectiveness of human performance. (Bennett – 1979) 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.

Usability Assessment in the Design Process: Usability assessment should begin in the early stages of the product development and should be continually applied throughout the process. The assessment should include the user's entire experience, and all the product's important components.

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 asystem.
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.

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 termsof 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 usertraining?
  - 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?

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. No one person possesses all the skills to perform all the necessary tasks; the best that can be hoped for is that one person may possess a couple of skills. 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 specialist to define behavioral requirements and apply behavioral 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 team members in close proximity to one another.

Step 1: KYC: Know Your Client/User

To create a truly usable system, the designer must always do the following:

- Understand how people interact with computers.
- Understand the human characteristics important in design.
- Identify the user's level of knowledge and experience.
- Identify the characteristics of the user's needs, tasks, and jobs.
- Identify the user's psychological characteristics.
- Identify the user's physical characteristics.
- Employ recommended methods for gaining understanding of users.

Why People Have Trouble with Computers?

What makes a system difficult to use in the eyes of its user? Listed below are several contributing factors that apply to traditional business systems.

- Use of jargon.
- Non-obvious design.
- Fine distinctions.
- Disparity in problem-solving strategies.
- Design inconsistency.

Responses to Poor Design

Errors are a symptom of problems. The magnitude of errors in a computer- based system has been found to be as high as 46 percent for commands, tasks, or transactions. Errors, and other problems that befuddle one, lead to a variety of psychological and physical user responses.

Typical Psychological responses to poor design are:- Confusion, Annoyance, Frustration, Panic or stress,Boredom.

Similarly Physical responses are:- Abandonment of the system, Partial use of the system, Indirect use of the system,Modification of the task, Compensatory activity. Misuse of the system,Direct programming.

Important Human Characteristics in Design

1. 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 color, 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; sometimeswe 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 v/s 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.

2. Memory

- Memory is viewed as consisting of two components, long-term and short-term (orworking) 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 andcomplexity 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.

3. 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.

4. 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 distanceof 2.5 degrees from the point of eye fixation
- The eye's sensitivity increases for those characters closest to the fixation point (the–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.

5. 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 theeye 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.

6. 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.

7. 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 systema 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.

8. Movement Control

- 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, thefaster 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.

9. **Learning**
   - Learning, as has been said, is the process of encoding in long-term memory information
   - 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.

10. **Skill**
   - The goal of human performance is to perform skillfully. 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.

11. **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 taskor 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 Considerations in Design

The kinds of user/task characteristics that must be established are summarized intable.

KNOWLEDGE/EXPERIENCE

Computer Literacy	Highly technical or experienced, moderate computer experience, or none.
System Experience	High, moderate, or low knowledge of a particular system and its methods of interaction.
Application Experience	High, moderate, or low knowledge of similar systems.
Task Experience	Level of knowledge of job and job tasks.
Other Systems Use	Frequent or infrequent use of other systems in doingjob.
Education	High school, college, or advanced degree.
Reading Level	Less than 5th grade, 5th–12th, more than 12th grade.
Typing Skill	Expert (135 WPM), skilled (90 WPM), good (55 WPM), average (40 WPM), or "hunt and peck" (10 WPM).
Native Language or Culture	English, another, or several.

JOB/TASK/NEED

Type of System Use	Mandatory or discretionary use of the system.
Frequency of Use	Continual, frequent, occasional, or once-in-a-lifetime use of system.
Task or Need Importance	High, moderate, or low importance of the task being performed.

Task Structure	Repetitiveness or predictability of tasks being automated, high, moderate, or low.
Social Interactions	Verbal communication with another person required or not required.
Primary Training	Extensive or formal training, self-training through manuals, or no training.
Turnover Rate	High, moderate, or low turnover rate for jobholders.
Job Category	Executive, manager, professional, secretary, clerk.
Lifestyle	For Web e-commerce systems, includes hobbies,recreational pursuits, and economic status.

PSYCHOLOGICAL CHARACTERISTICS

Attitude	Positive, neutral, or negative feeling toward job or system.
Motivation	Low, moderate, or high due to interest or fear.
Patience	Patience or impatience expected in accomplishing goal.
Expectations	Kinds and reasonableness.
Stress Level	High, some, or no stress generally resulting from task performance.
Cognitive Style	Verbal or spatial, analytic or intuitive, concrete or abstract.

PHYSICAL CHARACTERISTICS

Age	Young, middle aged, or elderly.
Gender	Male or female.
Handedness	Left, right, or ambidextrous.
Disabilities	Blind, defective vision, deafness, motor handicap.

1. The User's Knowledge and Experience

The following kinds of knowledge and experiences should be identified.

Computer Literacy: Are the users highly technical such as programmers or experienced data entryclerks or vice versa?

System Experience: Novice v/s Experts

- Words to describe the new, relatively new, or infrequent user have included naive, casual, inexperienced, or novice. At the other end of the experience continuum lie terms such as experienced, full-time, frequent, power, or expert. In between these extremes is a wide range of intermediate or intermittent users.
- In business systems, novice users have been found to:
  - Depend on system features that assist recognition memory: menus, prompting information, and instructional and help screens.
  - Need restricted vocabularies, simple tasks, small numbers of possibilities, and very informative feedback.
  - View practice as an aid to moving up to expert status.
- Experts, on the other hand:
  - Rely upon free recall.
  - Expect rapid performance.
  - Need less informative feedback.
  - Seek efficiency by bypassing novice memory aids, reducing keystrokes, chunking and summarizing
- Novice users often have difficulties:
  - Dragging and double-clicking using the mouse. Distinguishing between double-clicks and two separate clicks is particularly confusing
  - In window management. That overlapping windows represent a three-dimensional space is not always realized. Hidden windows are assumed to be gone and no longer exist.

- In file management. The organization of files and folders nested more than two levels deep is difficult to understand. Structure is not as apparent as with physical files and folders.

- Experts possess the following traits:
  - They possess an integrated conceptual model of a system.
  - They possess knowledge that is ordered more abstractly and more procedurally.
  - They organize information more meaningfully, orienting it toward theirtask.
  - They structure information into more categories.
  - They are better at making inferences and relating new knowledge to their objectives and goals.
  - They pay less attention to low-level details.
  - They pay less attention to surface features of a system.
- Novices exhibit these characteristics:
  - They possess a fragmented conceptual model of a system.
  - They organize information less meaningfully, orienting it toward surface features of the system.
  - They structure information into fewer categories.
  - They have difficulty in generating inferences and relating new knowledgeto their objectives and goals.
  - They pay more attention to low-level details.
  - They pay more attention to surface features of the system.

Application Experience

- Have users worked with a similar application (for example, word processing, airline reservation, and so on)? Are they familiar with the basic application terms?Or does little or no application experience exist?

Task Experience

- Are users experienced with the task being automated? Or do users possess littleor no knowledge of the tasks the system will be performing?

Other System Use

- Will the user be using other systems while using the new system?

Education

- What is the general educational level of users? Do they generally have high school degrees, college degrees, or advanced degrees?

Reading Level

- For textual portions of the interface, the vocabulary and grammatical structuremust be at a level that is easily understood by the users.

Typing Skill

- Is the user a competent typist or of the hunt-and-peck variety? Is he or she familiar with the standard keyboard layout or other newer layouts?

Native Language and Culture

- Do the users speak English, another language, or several other languages? Willthe screens be in English or in another language? Other languages often impose different screen layout requirements.
- Are there cultural or ethnic differences between users?

2. The User's Tasks and Needs

- The user's tasks and needs are also important in design. The following should be determined:

Mandatory v/s Discretionary Use

- Users of the earliest computer systems were mandatory or nondiscretionary. That is, they required the computer to perform a task that, for all practical purposes, could be performed no other way.
- This newer kind of user is the office executive, manager, or other professional,whose computer use is completely discretionary.

Characteristics of mandatory use can be summarized as follows:

- The computer is used as part of employment.
- Time and effort in learning to use the computer are willingly invested.
- High motivation is often used to overcome low usability characteristics.
- The user may possess a technical background.
- The job may consist of a single task or function.

Common general characteristics of the discretionary user are as follows:

- Use of the computer or system is not absolutely necessary.
- Technical details are of no interest.
- Extra effort to use the system may not be invested.
- High motivation to use the system may not be exhibited.
- May be easily disenchanted.
- Voluntary use may have to be encouraged.
- Is from a heterogeneous culture.

Frequency of Use

- Is system use a continual, frequent, occasional, or once-in-a-lifetime experience? Frequencyof use affects both learning and memory.
- Occasional or infrequent users prefer ease of learning and remembering,

Task or Need Importance

- How important is the task or need for the user?
  - People are usually willing to spend more time learning something if it makes thetask being performed or need being fulfilled more efficient.

Task Structure

- How structured is the task being performed? Is it repetitive and predictable or notso?

Social Interactions

- Will the user, in the normal course of task performance, be engaged in a conversation with another person, such as a customer, while using the system? If so, design should not interfere with the social interaction.
- Neither the user nor the person to whom the user is talking must be distracted in any way by computer interaction requirements. The design must accommodate the social interaction.

Job Category

- In a business system, is the user an executive, manager, professional, secretary, orclerk? While job titles have no direct bearing on design per se, they do enable one to predict some job characteristics when little else is known about the user.
- For example, executives and managers are most often discretionary users, while clerks are most often mandatory ones.

The User's Psychological Characteristics

- A person's psychological characteristics also affect one's performance of tasks requiring motor, cognitive, or perceptual skills.

Attitude and Motivation

- Is the user's attitude toward the system positive, neutral, or negative? Is motivation high, moderate, or low?
- While all these feelings are not caused by, and cannot be controlled by, the designer, a positive attitude and motivation allows the user to concentrate on the productivity qualities of the system.

Patience

- Is the user patient or impatient?
  - They are exhibiting less tolerance for Web use learning requirements, slow response times, and inefficiencies in navigation and locating desired content.

Stress Level

- Will the user be subject to high levels of stress while using the system? Interacting with an angry boss, client, or customer, can greatly increase a person'sstress level.
- System navigation or screen content may have to be redesigned for extreme simplicity in situations that can become stressful.

Expectations

- What are user's expectations about the system or Web site? Are they realistic?
  - Is it important that the user's expectations be realized?

Cognitive Style

- People differ in how they think about and solve problems.
- Some people are better at verbal thinking, working more effectively with wordsand equations.
- Others are better at spatial reasoning—manipulating symbols, pictures, and images.
- Some people are analytic thinkers, systematically analyzing the facets of a problem.
- Others are intuitive, relying on rules of thumb, hunches, and educated guesses.
- Some people are more concrete in their thinking, others more abstract.

The User's Physical Characteristics

The physical characteristics of people can also greatly affect their performancewith a system.

Age: Are the users children, young adults, middle-aged, senior citizens, or veryelderly? Age can have an effect on both computer and system usage.

Young Adults v/s Older Adults

Young adults (aged 18–36), in comparison to older adults (aged 64–81)

- Use computers and ATMs more often.
- Read faster.
- Possess greater reading comprehension and working memory capacity.
- Possess faster choice reaction times.
- Possess higher perceptual speed scores.
- Complete a search task at a higher success rate.
- Use significantly less moves (clicks) to complete a search task.
- Are more likely to read a screen a line at a time.

Older adults, as compared to young adults:

- Are more educated.
- Possess higher vocabulary scores.
- Have more difficulty recalling previous moves and location of previously viewedinformation.
- Have more problems with tasks that require three or more moves (clicks).
- Are more likely to scroll a page at a time.
- Respond better to full pages rather than long continuous scrolled pages.

Hearing

- As people age, they require louder sounds to hear, a noticeable attribute in almost any everyday activity.

Age in Years	Sound Level in dB
25	57
45	65
65	74
85	85

Vision

- Older adults read prose text in smaller type fonts more slowly than younger adults (Charness and Dijkstra, 1999). For older adults they recommend:
  - 14-point type in 4-inch wide columns.
  - 12-point type in 3-inch wide columns.
- Ellis and Kurniawan (2000) recommend the following fonts for older users:
  - San serif (Arial, Helvetica, and Verdana).
  - Black type on a white background.
- Ellis and Kurniawan (2000) and Czaja (1997) suggest Web links should be:
  - Distinct and easy to see.
  - Large (at least 180 × 22 pixels for a button).
  - Surrounded by a large amount of white space.

Cognitive Processing

- Brain processing also appears to slow with age. Working memory, attention capacity, and visual search appear to be degraded.
  - Tasks where knowledge is important show the smallest age effect and tasks dependent upon speed show the largest effect

Gender

- A user's sex may have an impact on both motor and cognitive performancebecause
  - Women are not as strong as men,
  - Women also have smaller hands than men, and
  - Significantly more men are color-blind than women

Handedness

- A user's handedness, left or right, can affect ease of use of an input mechanism, depending on whether it has been optimized for one or the other hand.

Disabilities

- Blindness, defective vision, color-blindness, poor hearing, deafness, and motor handicaps can affect performance on a system not designed with these disabilities in mind.
- People with special needs must be considered in design especially for systems likeweb design.

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 in below table.

Reading:	
Prose text:	250–300 words per minute.
Proof reading text on paper:	200 words per minute.
Proof reading 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:	
Typewriter	
Fast typist:	150 words per minute and higher.
Average typist:	60–70 words per minute.
Computer	
Transcription:	33 words per minute.
Composition:	19 words per minute.
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.

Methods for Gaining an Understanding of Users

- Gould (1988) suggests using the following kinds of techniques to gain an understanding of users, their tasks and needs, the organization where they work, and the environment where the system may be used.
    - Visit user locations, particularly if they are unfamiliar to you, to gain an understanding of the user's work environment.
    - Talk with users about their problems, difficulties, wishes, and what works well now. Establish direct contact; avoid relying on intermediaries.
    - Observe users working or performing a task to see what they do, their difficulties, and their problems.
    - Videotape users working or performing a task to illustrate and study problems and difficulties.
    - Learn about the work organization where the system may be installed.
    - Have users think aloud as they do something to uncover details that maynot otherwise be solicited.
    - Try the job yourself. It may expose difficulties that are not known, or expressed, by users.
    - Prepare surveys and questionnaires to obtain a larger sample of useropinions.
    - Establish testable behavioral target goals to give management a measurefor what progress has been made and what is still required.

Step 2: 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 isan 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 requirements.

DIRECT METHODS

Advantages

- The significant advantage of the direct methods is the opportunity they provide tohear the user's comments in person and firsthand.
- Person-to-person encounters permit multiple channels of communication (bodylanguage, 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 orsomewhat open-ended.
- A formal questionnaire should not be used, however. Useful topics to ask the userto 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.
- **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 willderive 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 techniquesshould 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 forvery 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 therequirements.
- The purpose of a focus group is to probe user's experiences, attitudes, beliefs, anddesires, 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 issuesand 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 comparativedata from several users.

Card Sorting for Web Sites

- A technique to establish groupings of information for Web sites.
  - Briefly, the process is as follows:
    - From previous analyses, identify about 50 content topics and inscribethem 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 colored 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 colored 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 orunintentional.
- Next, the intermediary may not possess a complete, or current, understanding ofuser'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, orhe or she may be from the using department.

Paper Survey or Questionnaire

- A survey or questionnaire is administered to a sample of users using traditionalmail 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 analyze.
- Questionnaires should be composed mostly of closed questions
- Questionnaires should be relatively short and created by someone experienced intheir design.

Electronic Survey or Questionnaire

- A survey or questionnaire is administered to a sample of users using e-mail or theWeb 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 usingworkstations.
- **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
- **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 analyzed (Customer Support, Technical Support, Help Desk, etc.).

E-Mail or Bulletin Board

- Problems, questions, and suggestions from users posted to a bulletin board or through e-mail are analyzed.

User Group

- Improvements are suggested by customer groups who convene periodically todiscuss software usage. They require careful planning.

Competitor Analyses

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

Trade Show

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

Other Media Analysis

- An analysis of how other media, print or broadcast, present the process, information, or subject matter of interest.

System Testing

- New requirements and feedback are obtained from ongoing product testing.

Requirements Collection Guidelines

- Establish 4 to 6 different developer-user links.
- Provide most reliance on direct links.

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 willbe 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 theusers 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—normallyorganization 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 color, 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 modelsevolve to become more accurate and complete. Making invisible parts of a system visible will speed up the process of developing correct mental models.

- **Provide proper and correct feedback:** Be generous in providing feedback. Keepa person informed of what is happening, and what has happened, at all times, including:
  - Provide visible results of actions.
  - Display actions in progress.
  - Provide a continuous indication of status.
  - Present as much context information as possible.
  - Provide clear, constructive, and correct error messages.
- **Avoid anything unnecessary or irrelevant:** Never display irrelevant informationon 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 conceptsto 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 behavior 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.

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.

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.

System Training and Documentation Needs

Training and documentation are also an integral part of any development process.

Training

- System training will be based on user needs, system conceptual design, systemlearning goals, and system performance goals.
- Training may include such tools as formal or video training, manuals, onlinetutorials, reference manuals, quick reference guides, and online help.
- Any potential problems can also be identified and addressed earlier in the design process, reducing later problems and modification costs.

Documentation

- System documentation is a reference point, a form of communication, and a more concrete design—words that can be seen and understood based on user needs, system conceptual design, and system performance goals.
    - It will also be Creating documentation during the development progress willuncover issues and reveal omissions that might not otherwise be detected until later in the design process.

USER INTERFACE DESIGN (18CS734)

SEMESTER – VII

Module – III

System menus and navigation schemes – Structures of menus, Functions of menus, Content of menus, Formatting of menus, Phrasing the menu, Selecting menu choices, Navigating menus, Kinds of graphical menus.

Step 4: Develop System Menus and Navigation Schemes

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

- The structures of menus.
- The functions of menus.
- The content of menus.
- Formatting menus.
- Writing menus.
- Navigation using menus.
- Web site navigation and links.
- Types of graphical menus.

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

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

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

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

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

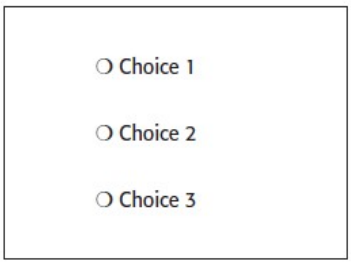


Figure 4.1 Single menu.

Sequential Linear Menus

- Sequential linear menus are presented on a series of screens possessing only one path.
- The menu screens are presented in a preset order, and, generally, their objective is for specifying parameters or for entering data.
- Sequential path menus have several shortcomings. A long sequence may become tedious as menu after menu is presented.

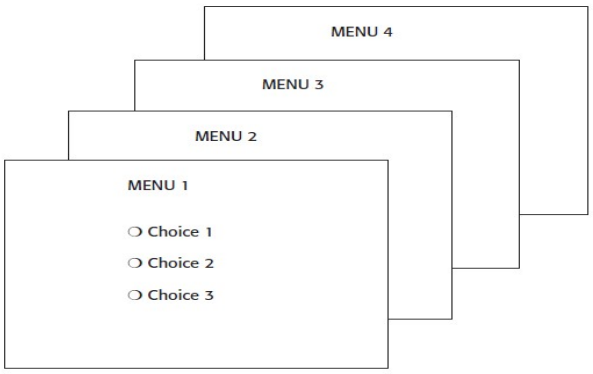
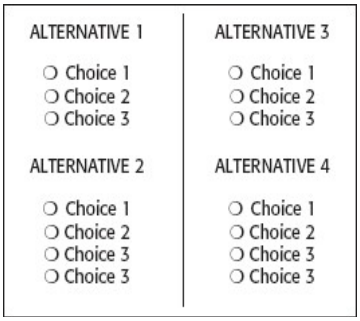


Figure 4.2 Sequential linear menus.

Simultaneous Menus

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



- 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 suboptions, from categories to subcategories, from pages to sections to subsections, and so on
- A hierarchical structure can best be represented as an inverse tree, leading to more and more branches as one moves downward through it.
- Common examples of hierarchical design today are found in menu bars with their associated pull-downs.
- A hierarchical structure is characterized by depth & breadth. *Depth* refers to number of choice levels one must traverse to reach the destination/target and *breadth* refers to number of alternatives found at each level.

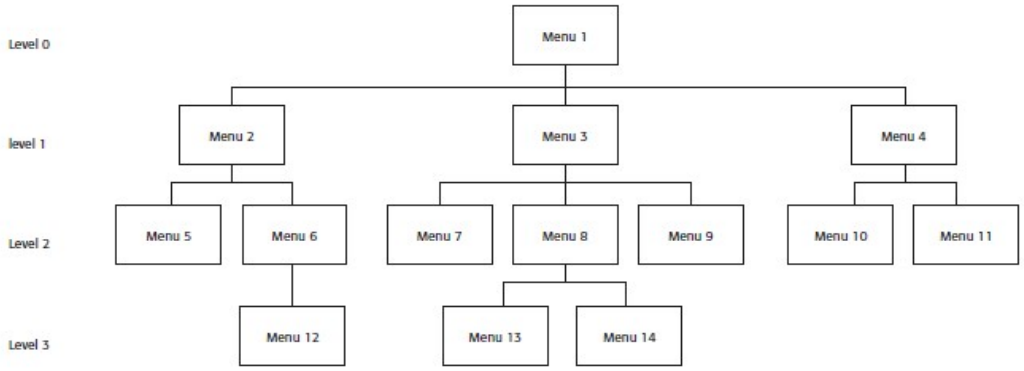


Figure 4.4 Hierarchical menus.

- A disadvantage of a hierarchical scheme is that the defined branching order maynot fit the users conception of the task flow.
- If users are not familiar with the hierarchical menu, or are unable to predict what suboptions lie below a particular choice, they may go down wrong paths and find it necessary to goback up the tree to change a choice, or perhaps even return to the top-level menu.

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.

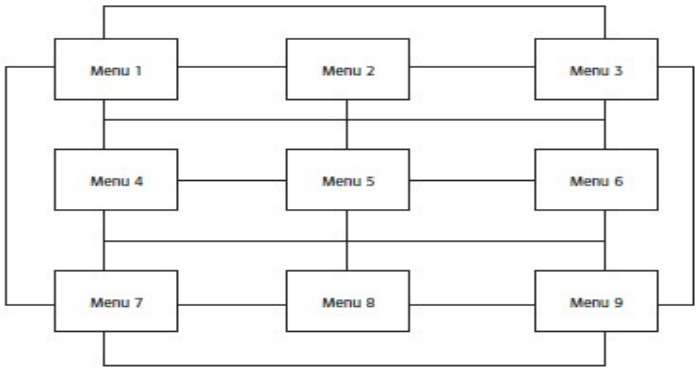


Figure 4.5 Connected menus.

- The biggest advantage of a connected menu network is that it gives the user fullcontrol over the navigation flow. Its disadvantage is its complexity,

Event-Trapping Menus

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

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

Functions of Menus

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

Navigation to a New Menu

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

Execute an Action or Procedure

- A user selection directs the computer to implement an action or perform aprocedure.
- 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 displayinformation.
- 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 nondestructive and usually undoable.

Data or Parameter Input

- Each selection specifies a piece of input data for the system or provides a parameter value. Data or values may be input on a single menu or spread over a hierarchy of menus.

Content of Menus

- A menu consists of four elements, its *context*, its *title*, its *choice descriptions*, and its *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 listingof 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 beconsidered in creating this kind of on-screen guidance.

Formatting of Menus

- HCI has a rich history of experimental studies with menus. The following is a series of guidelines for formatting menus.

Consistency

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

Display

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

Presentation

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

Organization

- Provide a general or main menu.
- Display:
  - — All relevant alternatives.
  - — Only relevant alternatives.
  - — Delete or gray-out inactive choices
- Match the menu structure to the structure of the task.
  - — Organization should reflect the most efficient sequence of steps to accomplisha 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 organizing menus the goal is to simply and effectively reveal its structure while reducing the number of actions needed to locate the target item.
- 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:
  - o Fewer steps and shorter time to reach one's objective.
  - o Fewer opportunities to wander down wrong paths.
  - o Easier learning by allowing the user to see relationships of menu items.
- A broad menu's disadvantages are:
  - o A more crowded menu that may reduce the clarity of the wording of choices.
  - o Increased likelihood of confusing similar choices because they are seen together.
- The advantages of greater depth are:
  - o Less crowding on the menu.
  - o Fewer choices to be scanned.
  - o Easier hiding of inappropriate choices.
  - o Less likelihood of confusing similar choices since there is less likelihood that they will be seen together.
- Greater depth disadvantages are:
  - o More steps and longer time to reach one's objective.
  - o More difficulties in learning since relationships between elements cannot always be seen.
  - o More difficulties in predicting what lies below, resulting in increasedlikelihood of going down wrong paths or getting lost.
  - o Higher error rates.

Complexity

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

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.

Ordering

- Order lists of choices by their natural order, or
- For lists associated with numbers, use numeric order.
- For textual lists with a small number of options (seven or less), order by:
  - — Sequence of occurrence.
  - — Frequency of occurrence.
  - — Importance.
  - — Semantic similarity.
- Use alphabetic order for:
  - — Long lists (eight or more options).
  - — Short lists with no obvious pattern or frequency.
- Separate potentially destructive actions from frequently chosen items.
- If option usage changes, do not reorder menus.
- Maintain a consistent ordering of options on all related menus.
  - — For variable-length menus, maintain consistent relative positions.
  - — For fixed-length menus, maintain consistent absolute positions.
- A meaningful ordering is necessary to:
  - o Facilitate search for an item.
  - o Provide information about the structure and relationships among items.
  - o Provide compatibility with the user's mental model of the item structure.
  - o Enhance the user's ability to anticipate a choice's location.

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.

Line Separators

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

Phrasing [Writing] 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.

The following guidelines are used for Phrasing [Writing] the Menu.

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.

Menu Choice Descriptions

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

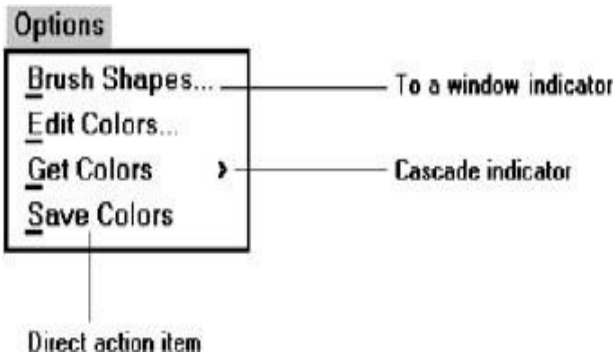
Menu Instructions

- 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 choicedescriptions 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 color.

The following standard graphical system conventions are inscribed on the menus.

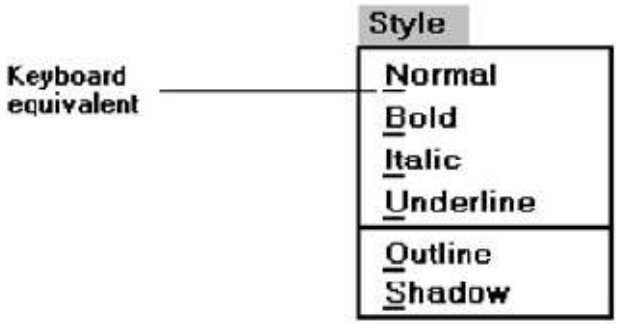
Intent Indicators

- Cascade indicator:
  - — To indicate that selection of an item will lead to a submenu, place a triangle or right-pointing solid arrow following the choice.
  - — A cascade indicator must designate every cascaded menu.
- To a window indicator:
  - — For choices that result in displaying a window to collect more information,place an ellipsis (. . .) immediately following the choice.
    - Exceptions—do not use when an action:
      - – Causes a warning window to be displayed.
      - – May or may not lead to a window.
- Direct action items:
  - — For choices that directly perform an action, no special indicator should beplaced on the menu.



Keyboard Equivalents

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



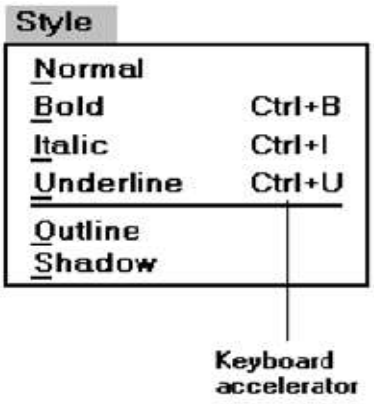
Standard Keyboard Equivalents

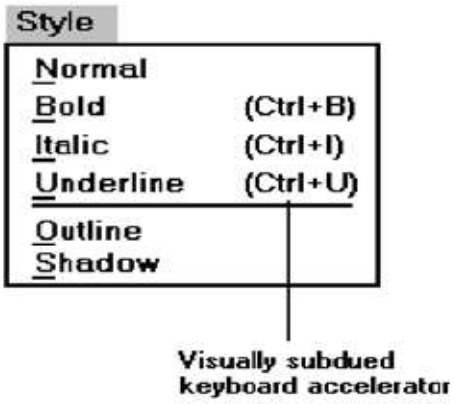
About	Help	Print	Send To
Apply	Help Topics	Print Preview	Show
Back	Insert	Properties	Size
Browse	Maximize	Redo	Split
Close	Minimize	Repeat	Stop
Copy	Move	Restore	Undo
Cut	New	Resume	View
Delete	Next	Retry	Yes
Edit	No	Rerun	
Exit	Open	Save	
File	Paste	Save As	
Find	Page Setup	Select All	

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.

Keyboard Accelerators

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





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

Initial Cursor Positioning

- If one option has a significantly higher probability of selection, position the cursorat 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.

Choice Selection

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

- Selection/execution:
  - — Provide separate actions for selecting and executing menu options.
  - — Indicate the selected choice through either:
    - Highlighting it with a distinctive display technique.
    - Modifying the shape of the cursor.
  - — Permit unselecting choice before execution.
    - If a menu is multiple choice, permit all options to be selected before execution.
- Combining techniques:

— Permit alternative selection techniques, to provide flexibility.

Defaults

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

Unavailable Choices

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

Mark Toggles or Settings

- Purpose:

— Use to designate that an item or feature is active or inactive over a relativelylong period of time.

— Use to provide a reminder that an item or feature is active or inactive.

- Guidelines:

— Position the indicator directly to the left of the option.

— For situations where several nonexclusive choices may be selected, consider including one alternative that deselects all the items and reverts the state to the −normal condition.

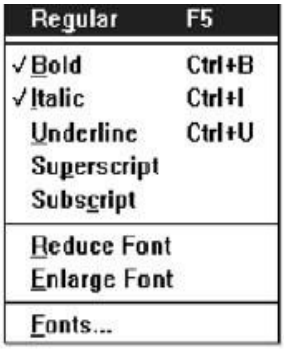


Figure 4.11 Mark toggles.

Toggled Menu Items

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

Kinds of Graphical Menus

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

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 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.

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.

Location

- Position choices horizontally over the entire row at the top of the screen, just below the screen title.
- A large number of choices may necessitate display over two rows.

Title

- The window title will be the menu bar title.

Item Descriptions

- The menu item descriptions must clearly reflect the kinds of choices available inthe 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.
- Do not display choices that are never available to the user.

Organization

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

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.

Separation

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

Other Components

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

Selection Indication

- Keyboard cursor:
  — Use a reverse video, or reverse color, 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 color, to highlight the selected choice.

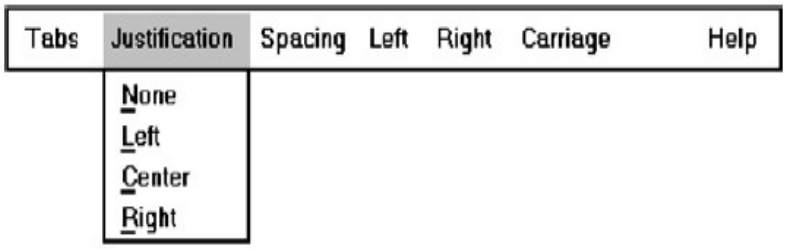
Pull-Down Menu

- 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.

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

- The disadvantages of pull-down menus are:
  - They require searching and selecting from another menu before seeing options.
  - They require looking away from main working area to read.

- They require moving the pointer out of working area to select (unless using keyboard equivalents).
- The items are smaller than full-size buttons, slowing selection time.
- The may obscure the screen working area.



Display

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

Location

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

Size

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

Title

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

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 colors, 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.

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.

Layout

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

Groupings

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

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.

Pull-Downs Leading to Another Pull-Down

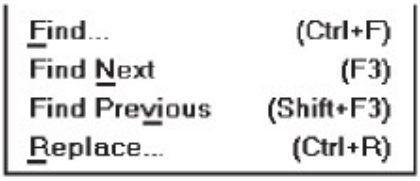
- If a pull-down choice leads to another pull-down, provide a cascade indicator as follows:
  — Place an arrow or right-pointing triangle after the choice description.
  — Align the triangles to the right side of the pull-down.
  — Display the triangle in the same color as the choice descriptions.

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 color as the choice descriptions.

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 color as the choice descriptions.



Separation

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

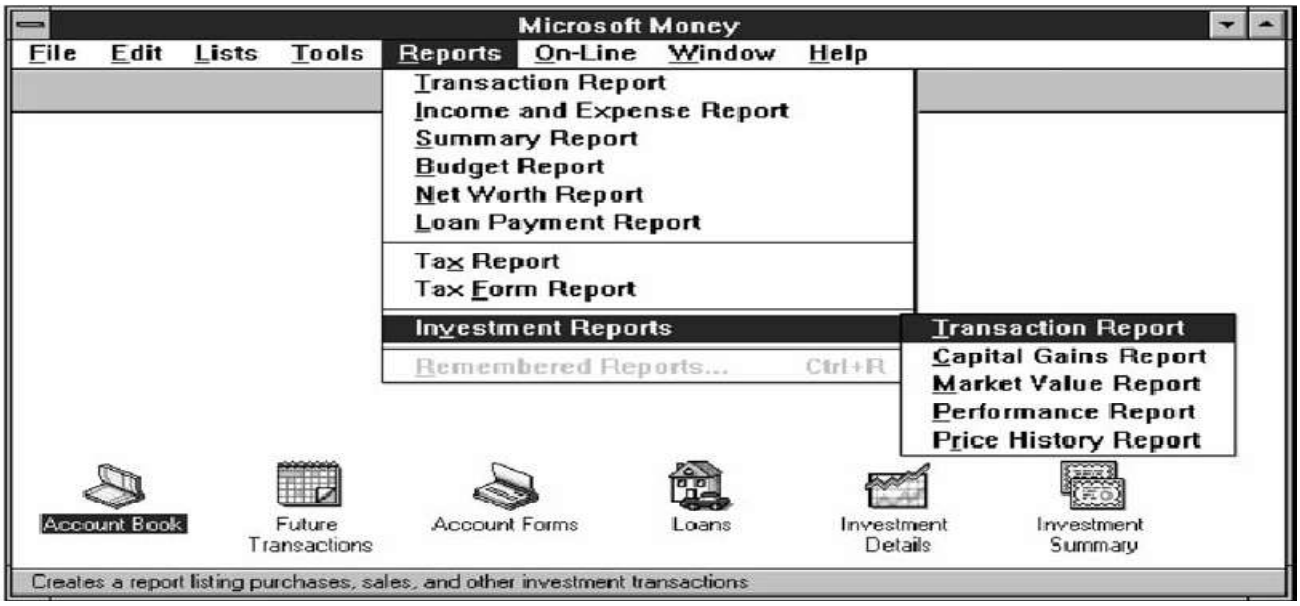
Selection Cursor

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

Cascading Menus

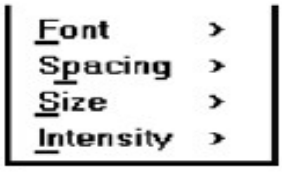
- Proper usage:
  — To reduce the number of choices presented together for selection (reduce menu breadth).
  — When a menu specifies many alternatives and the alternatives can be grouped in meaningful related sets on a lower-level menu.
  — When a choice leads to a short, fixed list of single-choice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



Cascade Indicator

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



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.

Levels

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

Title

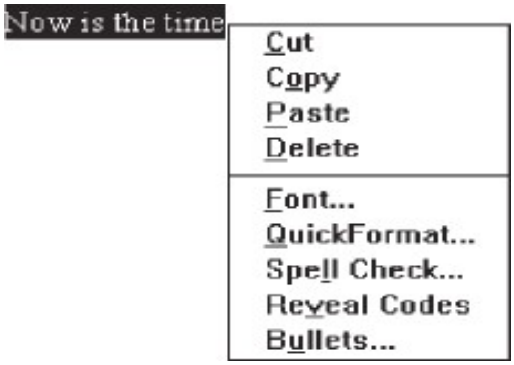
- Not necessary on the cascading menu.
  — The title will be the name of the higher-level menu item chosen.

Other Guidelines

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

Pop-up Menus

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



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.

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.

Size

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

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.

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.

Tear-off Menus

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

Iconic Menus

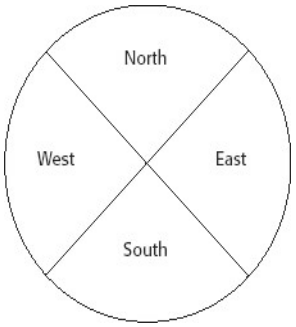
- Use to remind users of the functions, commands, attributes, or application choices available.
- Create icons that:
  — Help enhance recognition and hasten option selection.
  — Are concrete and meaningful.
  — Clearly represent choices.

Advantages/disadvantages.

- Pictures help facilitate memory of applications, and their larger size increases speed of selection. Pictures do, however, consume considerably more screenspace 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.

Pie Menus

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



Menu Proper Usage Summary

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.

USER INTERFACE DESIGN (18CS734)

SEMESTER – VII

Module – IV

Windows – Charactristics, Components of window, Window presentation styles, Types of window, Window management, Organizing window functions, Window operations, Web systems, Characteristics of device based controls.

Step 5: Select the Proper Kinds of Windows

- 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.
- A window may be small, containing a short message or a single field, or it may be large, consuming most or all of the available display space.
- A display may contain one, two, or more windows within its boundaries.

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.)
- Alocation, 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.

- **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.
- **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 another window stock maintenance can be done at the same time. Significant windows could remain displayed so that datials may be modified as needed prior to order completion.
- **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.
- **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, OS, applications, files, or areas of the same file. It may be presented on the screen alongside the problem, greatly facilitating its solution.

- **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.
- **Performing More Than One Task**: While waiting for along, complex procedure to finish, another can be performed. Tasks of higher priority can interrupt less important ones and then the interrupted tasks can be resumed without the necessity to ***close down*** and ***restart***.
- **Reminding**: Windows can be used to provide remainder through messages or popup or menus.
- **Monitoring**: Changes, both internal and external, can be monitored. Data in one window can be modified and its effect on data in another window can be studied.
- **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

Windowing systems, in spite of their appeal and obvious benefits, have failed to completely live up to their expectations. The problems with windowing systems can be attributed to the following three factors.

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

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 theycan be negated by excessive window manipulation requirements.
- 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

A typical window may be composed of up to a dozen or so elements. Some appear on all windows; others only on certain kinds of windows, or under certain conditions. For consistency purposes, these elements should always be located in the same position within a window. Most windowing systems provide consistent locations for elements in their own windows. What follows is a description of typical window components and their purposes, with emphasis on the most popular windowing system, **Microsoft Windows.** Specifically reviewed will be primary windows, secondary windows, and a form of secondary window called the dialog box. An illustration of a primary window is found in below figure. Illustratins of secondary windows and dialog boxes are illustrated later.

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.

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.

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.

- When these buttons are displayed, use the following guidelines:

- When a window does not support a command, donot display itscommand 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 b*utton 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.

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. Theymaybe located at the top ofthe screen in some platforms andcalled 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*.

Scroll Bars

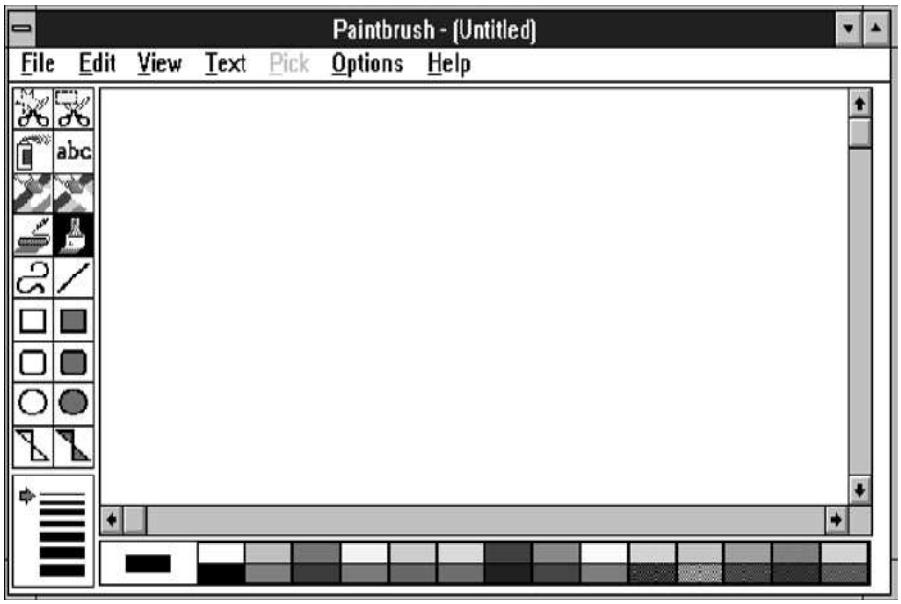
- When all display information cannot be presented in a window, the additional information must be found and made visible.
- This is accomplished by scrolling the display's contents through use of a scroll bar.
- A scroll bar is an elongated rectangular container consisting of a scroll area or shaft, 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 area, extending its entire length.

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 some times referred to as a *splitbar*.
- A window can be split into two or more separate viewing areas that are called *panes.*

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*.



Command Area

- Insituations 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.

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*.

Summary of window components for these windows is shown in he below table.

Table 5.1 Microsoft Windows Components

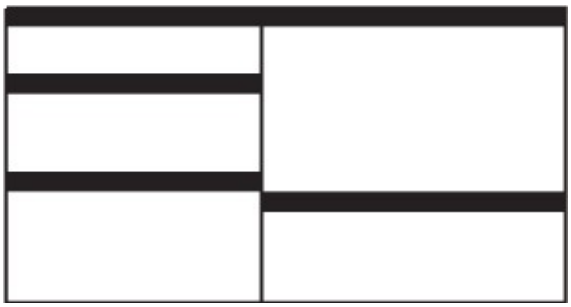
COMPONENT	WINDOWS CONTAINING COMPONENT		
	PRIMARY	SECONDARY	DIALOG BOX
Frame or Border • Boundary to define shape. • If sizable, contains control points for resizing.	X	X	X
Title Bar Text • Name of object being viewed in window. • Control point for moving window.	X	X	X
Title Bar Icon • Small version 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.

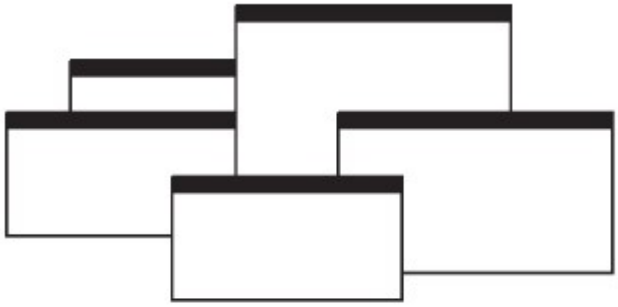


- **Advantages:**
  - The system usually allocates and positions windows for the user, eliminating the necessity to make positioning decisions.
  - Open windows are always visible, eliminating the possibility of them being lost and forgotten.
  - Every window is always completely visible, eliminating the possibility of information being hidden.
  - They are perceived as less complex than overlapping windows, possibly because there are fewer management operations or they seemless '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 numberof 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

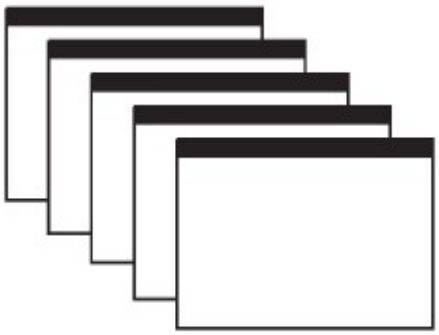
- Overlapping windows may be placed on top of one another like papers on a desk.
- They possess a 3-D quality, appearing to lie on different planes.



- **Advantages:**
  - Visually, their look is 3-D, 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.
  - 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 the below figure.



- **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

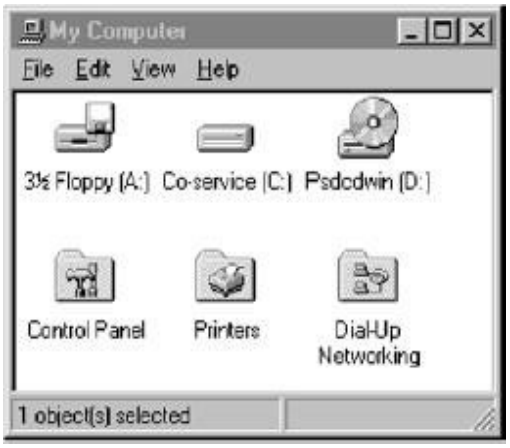
- Use tiled windows for:
  - Single-task activities.
  - Data that needs to be seen simultaneously.
  - Tasks requiring little window manipulation.
  - Novice or inexperienced users.
- Use overlapping windows for:
  - Switching between tasks.
  - Tasks necessitating a greater amount of window manipulation.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

Types of Windows

People's task must be structured into a series of windows. The type of window used will depend on the nature and flow of the task. Defining standard window type is again difficult across platforms because of varying terminology and definitions used by different windowing sytems. For simplicity, the Microsoft Windows windowing scheme is described with its windows, purpose and usage.

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.

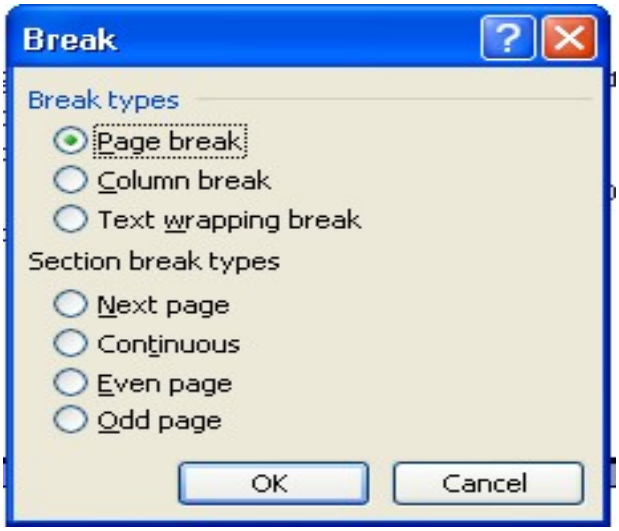


Proper usage:
— Should represent an independent function or application.
— Use to present constantly used window components and controls.
  • Menu bar items that are:
    — Usedfrequently.
    — Used bymost, or all, primary or secondary windows.
  • Controls used by dependent windows.
— Use for presenting 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.

Secondary Windows

Secondary windows are supplemental windows. Seconadry windows may be dependent upon a primary window or may be 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. Itis typically associated with a single data object, and appears on top of the active window whenrequested. 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.



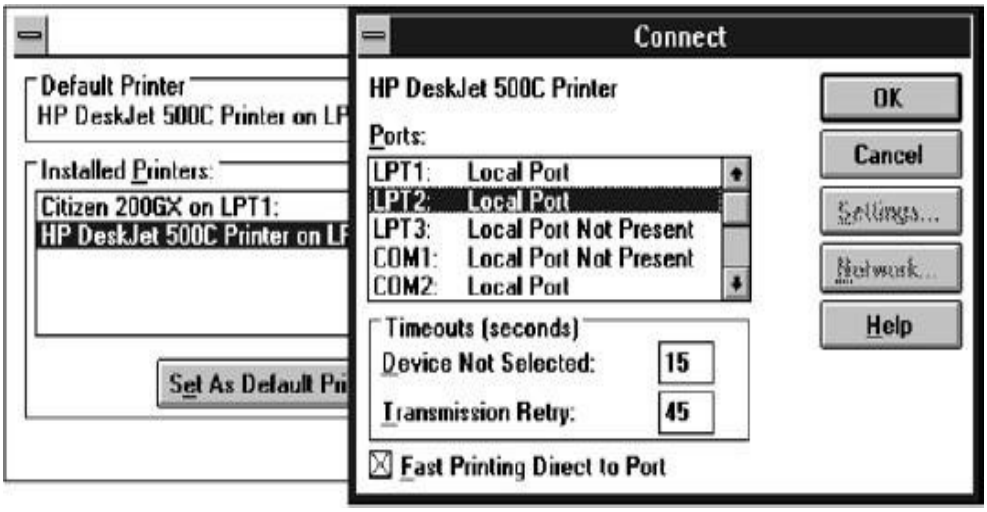
- Proper usage:
  — For performing subordinate, supplemental, or ancillary actions that are:
    • Extended or more complex in nature.
    • Related to objects in the primary window.
  — For presenting frequently or occasionally used window components.
- Important guidelines:
  — Should typically not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

Modal and Modeless

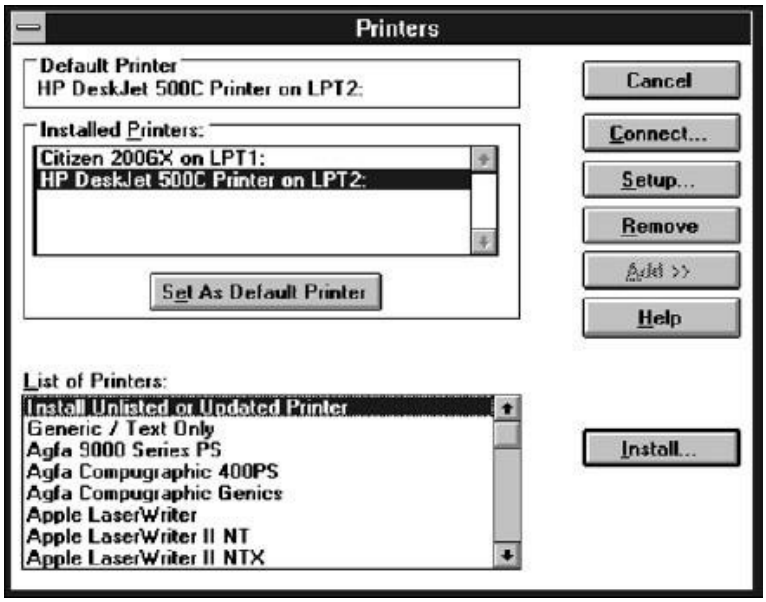
- Modal:
  — Use when interaction with any other window must not be permitted.
  — Use for:
    • Presenting information.
      — Forexample, messages (sometimes called a message box).
    • Receiving use rinput.
      — For example, data or information (sometimes called a prompt box).
    • Asking questions.
      — 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.

Cascading and Unfolding

- Cascading:
  — 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.

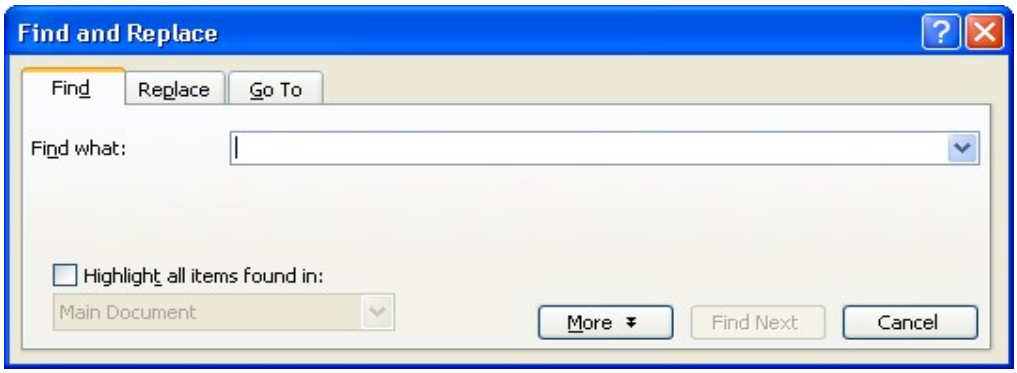


Cascaded Window



UnfoldedWindow

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.

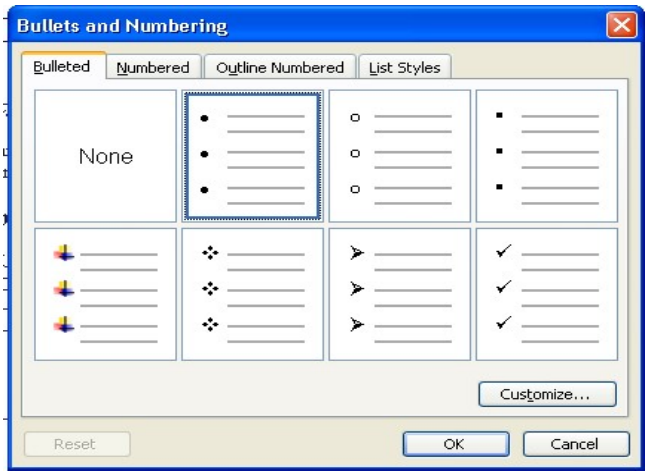
- Command buttons to include:

— OK.
— Cancel.
— Others as necessary.

Property Sheets and Property Inspectors

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 anobject.
- 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.
— Forsingle property sheets, place the commands on the sheet.
— For tabbed property pages, place the commands outside the tabbed pages.

Property Inspectors

- Usefordisplayingonlythe mostcommon orfrequentlyaccessed objectsproperties.
- Makechanges dynamically.

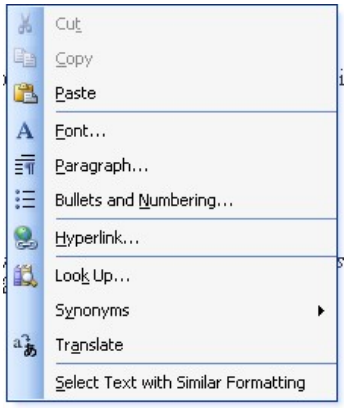
Message Boxes

- Usef or displaying a message a bout 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.
- 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 button.

Palette Windows

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

Pop-up Windows



- Use pop-up windows to display:
  — Additional information when an abbreviated form of the information is the main presentation.
  — Textual labels forgraphical controls.
  — Context-sensitive Help information

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.

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.

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.

Window Management

Microsoft Windows also provides several window management schemes, a *single document interface, a multiple-document interface, workbooks*, and *projects*. These are not exclusive design techniques. They may be combined or others may be developed. These techniques are the most frequently used schemes.

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.
- 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 in convenient 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 behavior are similar to those of the multiple document interface with all child windows maximized.
- Prope rusage:
  — To managea 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 simplicityof 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 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 menubar 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.
- Prope rusage:
  — 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

Information and functions must be presented to people when and where they need them. Proper organization and support of tasks by windows will only be derived through a thorough and clear analysis of tasks.

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

Windows are a means to end, a method of accomplishing something. Multiple windows on a display can be confusing, can increase load on the human visual system.

- The general rules are :
  - Minimize the number of windows needed 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

Guidelines for windows operations are still evolving. Guidelines will continue to develop and change as our understanding of, and experiences with, the windows interface continue to increase. Today, the following guidelines seem appropriate.

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 settingup 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 stateas 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 7 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 anypoint 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/Mouseless 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 the limited window 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.

Popup Windows:

JavaScript pop-up windows began appearing on the Web in 1996. Their use is multiplying and, in the view of almost all Web users, polluting screens. Because they are most frequently used in advertising, they have become a source of great aggravation to almost every user. Anecdotal evidence suggests that when a pop-up window begins to appear, most people close them before they are rendered. So, if a pop-up window is used, it may never be completely seen or read by the user. Use them with extreme caution.

Step 6: 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. The evolution of graphical systems has seen a whole new family of devices provided to assist and enhance this communication. These new machanisms are most commonly referred to as pointing devices.

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 lightpen, the finger, and voice.
- *Indirect* devices are operated in a location other than the screen, most often on the desktop. Examples include the mouse, trackball, and keyboard.
- The psychometer skills involved in learning to use, and using, a direct device are much simpler than those required for an indirect device. Most of these direct device skills were instilled in our formative years.

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 interms 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 interms 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, lightpen, 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 froms creen.
  — 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 at least ¾" 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 tobegin using.

LightPen

- 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 interms of direction, distance, and speed.
  — Movement is direct, in the same plane as screen.
  — Requires minimal additional desk space.
  — Stands up well in high-useenvironments.
  — More accurate than finger touching.
- Disadvantages:
  — Hand may obscure part of screen.
  — Requires picking it upto 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 might require extremely fine mouse positioning.
  - Never use double-clicks or double-drags as the only means of carrying out essential operations.
  - Do not use mouse plus keystroke combinations.
  - Do not require a person to point at a moving target.

Keyboard

- Description:
  — Standard typewriter keyboard and cursor movement keys.
- Advantages:
  — Familiar.
  — Accurate.
  — Does not take up additional deskspace.
  — 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.

==========

Step 14: Test, Test, and Retest

Once a system has been implemented, it is aggravating, costly & time consuming for both users & developers to uncover and correct any system usability deficiencies. After implementation many problems may never be corrected because of time constraints and cost. To minimize these kinds of problems and ensure usability, interfaces must be continually tested and refined before they are implemented.
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 alsocan 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 continuethroughout the development process.
- It should include as many of the user's tasks, and as many of the product'scomponents, 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.

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

Interactive Paper Prototypes

- Description:
— Interface components (menus, windows, and screens) constructed of commonpaper technologies (Post-It notes, transparencies, and so on).
— The components are manually manipulated to reflect the dynamics of thesoftware.
— 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.

Programmed Facades

o Description:
— Examples of finished dialogs and screens for some important aspects of thesystem.
— Created by prototyping tools.
— Medium-fidelity to high-fidelity prototypes.
o Advantages:
— Provide a good detailed specification for writing code.
— A vehicle for data collection.
o 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, schedulelimitations, or lack of user interest.
— Not practical for investigating more than two or three approaches.

Prototype-Oriented Languages

- Description:
— An example of finished dialogs and screens for some important aspects of thesystem.
— Created through programming languages like Power Builder, Visual Basic, that supports the actual programmingprocess.
— 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.

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.

Heuristic Evaluation

- Description:

— A detailed evaluation of a system by interface design specialists to identifyproblems.

- 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 usercommunities.
— 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**

o 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.

o 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.
o After the session:
— Hold a debriefing session including observers and design team memberswhere:
• 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 systemare 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 anddeficiencies.

- **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

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).

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.

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.

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.

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 analyzing 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.

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.

A final word

Application of guidelines and principles in design will aid greatly in creating a product that satisfies all the needs of clients. A happy and satisfied client, of course, also means a happy and satisfied developer.

==========